Journal of Language Modelling

VOLUME 13 ISSUE 1
JUNE 2025

Institute of Computer Science
Polish Academy of Sciences
Warsaw

Journal of Language Modelling

VOLUME 13 ISSUE 1
JUNE 2025

Articles

JOURNAL OF
LANGUAGE MODELLING

ISSN 2299-8470 (electronic version)
ISSN 2299-856X (printed version)
http://jlm.ipipan.waw.pl/

Published by IPI PAN
Institute of Computer Science, Polish Academy of Sciences
ul. Jana Kazimierza 5, 01-248 Warszawa, Poland

Circulation: 50 + print on demand

Layout designed by Adam Twardoch.
Typeset in XƎLaTeX using the typefaces: *Playfair*
by Claus Eggers Sørensen, *Charis SIL* by SIL International,
JLM monogram by Łukasz Dziedzic.

From speech signal to syntactic structure: A computational implementation

Tina Bögel and Tianyi Zhao
University of Konstanz

ABSTRACT

This paper presents a new computational implementation bridging several modules of grammar from phonetics to phonology to syntax. The system takes as input a speech signal annotated with syllables, interprets the phonetic data in phonological/prosodic terms, matches the data against a lexicon and makes the results available to a linguistically deep computational grammar. The system is showcased by means of syntactically ambiguous structures in German which can be disambiguated based on prosodic constituency information. A system evaluation with the German data showed good results for this new combination of automatic speech signal analysis and computational grammars, which takes a significant step towards a linguistically fine-grained computational analysis and hence towards real automatic speech understanding.

Keywords: Automatic speech understanding, syntactic ambiguities, prosodic disambiguation, LFG, German

INTRODUCTION 1

Spoken language is notoriously difficult for linguistic analyses in general and for computational implementations in particular. Various acoustic features such as duration, pitch contours, or voice quality contribute to the overall interpretation of an utterance, but are gradient in nature and subject to variation between and within speakers. This makes it very challenging for computationally deep linguistic grammars to use information signalled by prosodic structure. As a consequence, linguistically relevant information is often lost during analysis. Consider, for example, the following statement with contrastive focus on *red*.

(1) Amra ate the RED apple.

The contrastive focus in example (1) can be acoustically signalled by a strong tonal accent with a steep rise on *red* (e.g., Xu and Xu 2005; Gussenhoven 2008) which also has implications for the meaning interpretation of the clause: Not only did Amra eat a red apple, but she ate (for example) neither the green nor the yellow apple. These types of foci often correct wrong assumptions in the interlocutors' common ground and are thus highly relevant for analyses concerned with discourse or information structure (Krifka 2008; Rooth 2016).

Another common issue is the determination of prosodic constituency in the context of syntactic ambiguities as in example (2) where *flat* can be either associated with the preceding phrase (2a) or the following phrase (2b).

(2) a. When the cake was dropped flat || plants stuck to its underside.
 b. When the cake was dropped || flat plants stuck to its underside.

There are two possible syntactic analyses: a resultative structure as in example (2a) (... *drop the cake flat* ...), or a modifying structure as in example (2b) (... *flat plants* ...). Depending on whether the prosodic phrase boundary (||) precedes or follows the adjective *flat*, one of the interpretations becomes more likely (Bögel and Turk 2019). Such structures frequently appear in a variety of languages and it has been shown that many can be disambiguated by prosody (Lehiste *et al.* 1976; Price *et al.* 1991). Consequently, access to this information prevents overgeneration and supports meaning interpretation.

These are just two cases where prosodic information plays a crucial role in linguistic analyses, but numerous other examples can be found in a variety of linguistic structures across languages, e.g., the distinction between polar and constituent questions in Urdu by means of tonal accents (Butt *et al.* 2020), the second position placement of oblique pronoun clitics in Vafsi (Bögel *et al.* 2018), or the signalling of a rhetorical question by means of pitch contour, constituent duration, and voice quality in German (Braun *et al.* 2019). This shows that access to information from the speech signal, e.g., concerning pitch distribution and prosodic constituency, benefits speech recognition and interpretation and is thus very desirable for linguistically deep computational grammars.

However, an integration of prosodic information with existing grammars is rarely pursued, although several approaches supporting automatic speech recognition and the determination of prosodic events are available and are widely used in phonetic and prosodic research. The Munich automatic segmentation system MAUS (Kisler *et al.* 2017; Schiel 1999), for example, is frequently utilized to automatically annotate segments and words in more than 20 languages such as English, German, French, and Finnish, but does not include the calculation of pitch accents or prosodic constituency. By contrast, ProsodyPro (Xu 2013) is used to analyze speech prosody with both discrete and continuous data as output, with a focus on time-normalized pitch contours and F_0 velocity. F_0 contours and other acoustic cues can be averaged across repetitions and speakers, which enables a direct statistical comparison. However, the system does not provide any categorical information, e.g., in terms of accents, and calculates the data without the consideration of sentence, word, or syllable structures which makes it difficult to (re-)associate the output with, e.g., syntactic constituents.

There are several approaches to the automatic annotation of prosodic events with relation to corpora (often with a focus on future speech synthesis) that go beyond the sole interpretation of acoustic cues and include basic morphosyntactic information as well, e.g., in form of part-of-speech (POS) tags. The *Prosodizer* (Braunschweiler 2003, 2006) can assign pitch accents and boundary tones during speech recognition in American English and German speech corpora following the ToBI labelling conventions (Silverman *et al.* 1992). The method relies on acoustic features as well as syntactic boundary labels and POS tags which are part of the corpus annotations. An evaluation showed more than 70% accuracy in pitch accent and boundary tone detection with major difficulties at the level of intermediate phrase boundaries. The multilingual prosody module of the Verbmobil system integrates a word-based annotation and classification of boundaries, phrase accents, and sentence mood for German, English, and Japanese dialogues (Batliner *et al.* 2000, 2001; Wahlster 2013). Schweitzer and Möbius (2009) went beyond the word base and trained a number of classifiers on acoustic, phonological, and basic morphosyntactic attributes of German using the WEKA machine learning software (Witten and Frank 2005), reaching recognition accuracy rates of up to ~86% for the occurrence of accents, and ~93% for the occurrence of larger boundaries.

All of these approaches allow for the recognition and depiction of prosodic events in form of boundaries and accents, but none of them allow for real communication between prosodic structure and other modules of grammar. If (morpho)syntactic information in a given corpus is included in the system, it is used to facilitate prosodic annotation, but not vice versa, i.e., prosodic information is not used to determine (morpho)syntactic structure. None of the approaches are designed to allow for the prosodic disambiguation of syntactic structure or for signalling focus structures in order to enhance linguistic analyses by computational grammars.

Current large-scale grammar development projects which provide deep linguistic analyses include the Parallel Grammar project (ParGram, Butt *et al.* 2002; Sulger *et al.* 2013) based on Lexical-Functional Grammar (LFG; Kaplan and Bresnan 1982) and the DELPH-IN project in combination with the LinGO (Linguistic Grammars Online) Matrix effort (Bender *et al.* 2002; Copestake 2002) based on Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag 1994). Other major grammar development efforts are based on CCG (Steedman 2000; Clark and Curran 2007) and TAG (Joshi 2003; Duchier *et al.* 2004; Gardent and Parmentier 2005).

So far, these grammar development approaches have focussed on the syntactic and semantic representation of language. There are no detailed implementations of p-structure (including prosody and (post)lexical phonology), although some initial attempts restricted to specific phonological phenomena have been made across frameworks (see, for example, Butt and King 1998, Bird 1992, Bird and Klein 1994, Klein 2000). Computational approaches to specific/isolated phonological phenomena without integration into a large-scale grammar have also been developed in frameworks based on constraint rankings (as in Optimality Theory (Prince and Smolensky 2004); see, e.g., Tesar and Smolensky 1998; Becker *et al.* 2007; Yu 2018) and constraint weighting (as in Harmonic Grammar (Legendre *et al.* 1990); see, e.g., Potts *et al.* 2010). Penn and Carpenter (1999) combine two smaller-scale HPSG grammars of English and German with off-the-shelf speech recognition and TTS systems to allow for automatic translation and generation of spoken language. However, their system only includes spoken language in a detached manner in that a speech signal is first converted into a simple text string (which is then further processed by the grammar) and vice versa. To date, a real integration of spoken language into a large-scale computational grammar to enable deep automatic speech understanding has not been accomplished.

This paper uses the computational grammars developed in the spirit of Lexical-Functional Grammar (LFG), which have long been established as part of the ParGram project and have been used for a multitude of purposes with a strong focus on syntactic and semantic processing (a.o., Butt *et al.* 1999, 2002; Bobrow *et al.* 2007; Sulger *et al.* 2013; Crouch *et al.* 2017; Meßmer and Zymla 2018; Dalrymple *et al.* 2019). The input to all of these grammars is the s(yntactic)-string, which consists of a string of words that make up a written sentence (or a fragment thereof). In a standard computational LFG grammar, this string is tokenized into single words whose lexical morphosyntactic information is accessed and made available for further processing of the string in c(onstituent)- and f(unctional)-structures as well as semantic representations. This basic structure (including variations or extensions thereof) has been the established core structure of all computational LFG grammars. Grammars can be built via XLE, a state-of-the-art grammar development platform (a.o., Butt *et al.* 1999; Crouch *et al.* 2017), which allows researchers to build industrial-strength computational grammars for a wide range of languages and can be integrated with industrial-strength finite-state morphologies (Beesley and Karttunen 2003; Kaplan *et al.* 2004; Bögel *et al.* 2007).[1]

[1] See the XLE-Web interface which features a number of different computational LFG grammars that can be used interactively: https://clarino.uib.no/iness/xle-web.

While these grammars are well-established for syntactic and semantic analyses of texts, they are as of yet unable to process spoken language. As a consequence, linguistic phenomena whose analysis requires prosodic information (as demonstrated in examples (1) and (2)) cannot be interpreted by the traditional computational LFG grammars, although the combination of automatic speech recognition with linguistically deep computational grammars would be highly desirable and benefit both automatic speech understanding and speech synthesis.

This paper introduces a new system which bridges this gap between the automatic recognition of prosodic events and their linguistically deep analysis by computational LFG grammars, taking the prosodic disambiguation of syntactically ambiguous structures as a demonstration example. The implementation includes a representation of the speech signal in phonetic and phonological/prosodic terms, where the categorical representation of the latter enables the computational grammars to prosodically disambiguate syntactically ambiguous structures. This not only reduces overgeneration in the case at hand, but makes a linguistically fine-grained representation of prosodic categories (accents and boundaries) available for other modules of grammar, thus taking a huge step towards real automatic speech understanding.

The paper is structured as follows: Section 2 introduces the syntactically ambiguous data and briefly reports on a production experiment that establishes the relevant acoustic features for a prosodic disambiguation. Section 3 first gives a brief introduction to LFG and then describes the theoretical foundations behind the approach to the prosody-syntax interface proposed in this paper. Section 4 describes in detail all aspects of the computational implementation, from the interpretation of the speech signal to the disambiguation of syntactically ambiguous structures. This is followed by an evaluation of the system in Section 5. Section 6 concludes the paper.

THE DATA: SYNTACTICALLY AMBIGUOUS STRUCTURES 2

The following German example (3) has two possible interpretations:

(3) Sie sahen, dass
They saw that

[der Partner]$_{NP1}$ [der Freundin]$_{NP2}$ fehlte
the.MASC.NOM partner the.FEM.GEN/DAT friend was.missing

a) "They saw that the friend's partner was missing."

b) "They saw that the friend missed the partner."

There are two sources of ambiguity in this example: the syncretism of the determiner *der* 'the' and the verb's valency. The determiner is ambiguous in this position as it can be interpreted either as feminine dative or feminine genitive (Table 1), which makes the complete second NP *der Freundin* 'the friend' be interpreted as either dative or genitive. Adding to this local ambiguity is the valency of the verb *fehlen* 'missing', which can be used in either intransitive or transitive constructions, the latter requiring a dative object. As a result, the second NP can either be interpreted as a dative object to the verb or as a possessor phrase to the first NP *der Partner*, as indicated by the two translations given in example (3). Such syntactically ambiguous structures result in overgeneration, i.e., the (computational) grammar returns several possible solutions as illustrated in Figure 1. Previous research has shown that syntactically ambiguous structures can often be disambiguated by means of prosody (Price *et al.* 1991) and several studies have demonstrated this for a number of German structures as well (Żygis *et al.* 2019; Gollrad *et al.* 2010).

case	masc	fem	neut
gen	des	*der*	des
dat	dem	*der*	dem

Table 1:
The German determiner system for the singular genitive and dative

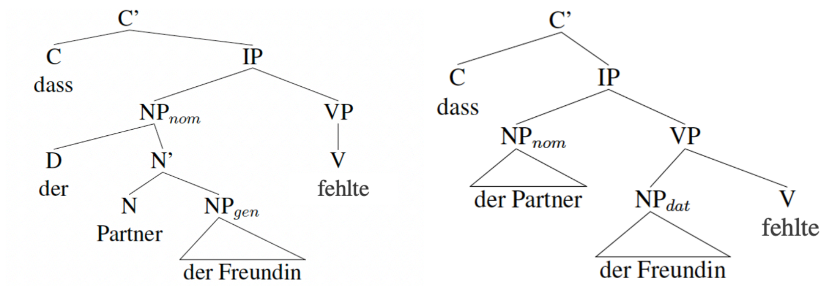


Figure 1: Two syntactic interpretations for example (3): genitive structure on the left, dative structure on the right

For structures as in example (3), current theories of the syntax-prosody interface would predict a prosodic phrase boundary to occur between the two NPs in the dative construction, but not in the genitive. Table 2 illustrates the predictions made by Selkirk's (2011) MATCH THEORY, which posits a phonological phrase (PhP/φ) for every syntactic XP (NP, PP, ...), in combination with Truckenbrodt's (1999) WRAP constraint, which assumes that a recursive XP/PhP is merged ('wrapped') into a single PhP.

For the syntactic structures given in Figure 1 and the string *der Partner der Freundin*, MATCH THEORY predicts a PhP boundary for every NP, resulting in two PhPs for the dative structure, and one nested PhP in the genitive structure. WRAP then assumes that the nested PhP in the genitive is wrapped into a single PhP. The algorithm thus assigns a PhP boundary after the first NP in the dative, but not in the genitive structure, as illustrated in Table 2.

Table 2: Prosodic phrasing predictions for the syntactic structures in Figure 1

Dative	Syntax		[der Partner]$_{NP}$ [der Freundin]$_{NP}$	
	Prosody	MATCH	$_{\varphi}$(der Partner)$_{\varphi}$(der Freundin)$_{\varphi}$	↕
		WRAP	$_{\varphi}$**(der Partner)**$_{\varphi}$**(der Freundin)**$_{\varphi}$	
Genitive	Syntax		[der Partner [der Freundin]$_{NP}$]$_{NP}$	
	Prosody	MATCH	$_{\varphi}$(der Partner $_{\varphi}$(der Freundin)$_{\varphi}$)$_{\varphi}$	↕
		WRAP	$_{\varphi}$**(der Partner der Freundin)**$_{\varphi}$	

In a production experiment, Bögel (2020) confirmed the theoretical predictions in Table 2. The stimuli consisted of nine fully ambiguous structures similar to example (3), where the first NP was always masculine and the second one feminine, followed by a verb with an ambiguous valency. All nouns had a disyllabic, trochaic foot structure (i.e., the first syllable carried lexical stress and the second one was unstressed (x -)).

The participants were fifteen female native speakers of German.[2] Each participant was presented with a context and a target sentence. Participants were asked to read the context silently and to 'mentally understand' the sentence before producing it as naturally as possible. Each participant produced 18 sentences (9 genitive and 9 dative constructions), resulting in a total of 270 sentences.

A linear mixed effects regression model (lmer) with items and subjects as random factors yielded the following results:

- A significantly steeper **drop in the fundamental frequency (F_0)** ('Reset') between NP1 and NP2 (as measured at the final syllable of NP1 and the determiner of NP2) in the dative as compared to the genitive condition ($\beta = -9.31$, SE $= 2.64$, t $= -3.53$, $p < 0.01$).
- A **pause**[3] between the first and the second NP in the dative condition: ($\beta = -2.35$, SE $= 0.92$, t $= -2.55$, $p < 0.05$).
- The **duration** of the last syllable of the first NP was significantly longer in the dative condition than in the genitive condition ($\beta = -2.8$, SE $= 0.79$, t $= -3.58$, $p < 0.01$).

These findings confirm the placement of a prosodic phrase boundary after the first NP in the dative, and provide detailed information on the relevant acoustic indicators of a prosodic phrase boundary, namely duration, F_0 movement, and pauses.

While the experimental results are in line with the predictions in Table 2, the question remains how these findings can be used to prosodically disambiguate syntactically ambiguous structures in LFG.

[2] The main goal of the original production experiment was to find the prosodic cues that disambiguate the syntactic structures. In order to reduce variation with respect to pitch evaluation, the decision was made to only record female participants. For the computational implementation described below this has no effect, since the implementation normalizes pitch by means of semitones.

[3] Following the MAUS conventions, a pause is defined as a silence interval which lasts more than 100 ms. See https://clarin.phonetik.uni-muenchen.de/BASWebServices/help/help_faq#help_faq.

3 THEORETICAL ANALYSIS IN LFG

After a brief introduction to LFG, this section discusses the architectural assumptions made with respect to the interface between syntax and prosody from a theoretical perspective which in turn forms the basis for the computational implementation in Section 4.

The generative, non-transformational LFG framework (Kaplan and Bresnan 1982; Bresnan *et al.* 2016; Börjars *et al.* 2019; Dalrymple *et al.* 2019; Dalrymple 2023) has a modular architecture with parallel representative structures for separate linguistic aspects which constrain each other through mathematically well-defined functions. Different types of linguistic information are encoded in suitable representation structures. For example, the original core structures c(onstituent)-structure and f(unctional)-structure both represent different aspects of syntactic structure: While c-structure depicts linear order and syntactic constituency by means of tree diagrams as in Figure 1, f-structure captures key dependency relations like grammatical functions (e.g., subject and object) as well as other functional information such as tense/aspect or case. F-structures are represented in Attribute-Value-Matrices (AVMs) and are largely invariant across languages. These two structures are linked via the projection function ϕ to allow for communication between syntactic constituency and related functional information. A number of additional structures have been proposed over the years, including a(rgument)-structure, i(nformation)-structure, and m(orphological)-structure, each of which represents the linguistic information associated with that aspect of grammar. Correspondence between these structures is again ensured via well-defined projection functions (see Dalrymple 2023 for a general introduction to LFG).

Several proposals have also been made for p(rosodic)-structure (see Bögel 2023 for an overview). This paper follows the proposal made in Bögel 2015. It distinguishes between *comprehension* ('parsing' in computational terms), which describes the processing and subsequent understanding of the speech signal by a listener, and *production* ('generation' in computational terms), which describes the process from the initial concept to the actual form of an utterance. The present paper focuses on comprehension: It discusses the process of going from a speech signal to a linguistic analysis, i.e., from phonetics to prosody to syntax.

In the proposal made by Bögel (2015), information at the prosody-syntax interface is exchanged on two levels: a) the *transfer of vocabulary* (ρ/π), which exchanges phonological and morphosyntactic information of lexical elements via a multidimensional lexicon, and b) the *transfer of structure* (♮), which exchanges information on syntactic and prosodic phrasing, and on intonation. Figure 2 illustrates this interaction in LFG where syntactic constituent structure is represented by c-structure, prosodic/phonological information by p-structure, and the s(yntactic)-string is placed between them. Mathematically well-defined projection functions (here: ♮, ρ, π) allow for the correspondence between these modules.

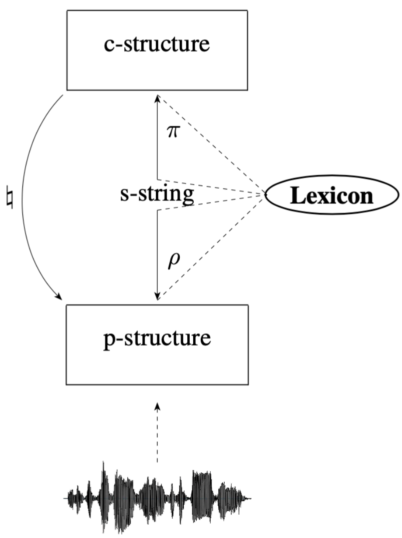


Figure 2:
The underlying architectural assumptions for the interface between syntax (c-structure) and prosody (p-structure)

P-structure 3.1

P-structure is represented via the p-diagram, a linear syllable-based representation of the speech signal over time (Figure 3). During comprehension, acoustic information from the speech signal feeds into p-structure and is stored at the *signal level*. Each syllable in the signal

...	...	...	...	...	...	...	signal
DURATION	0.15	0.25	0.25	0.13	0.31	0.19	↓
FUND. FREQ.	192	181	269	209	188	218	
SEGMENTS	[de:6]	[pa6t]	[n6]	[de:6]	[fROYn]	[dIn]	
VECTORINDEX	**S_1**	**S_2**	**S_3**	**S_4**	**S_5**	**S_6**	

Figure 3: The signal level of the p-diagram for *der Partner der Freundin*

receives a vector (S_n) which contains information, e.g., on the segments,[4] the duration, or the mean fundamental frequency (F_0) of that syllable.[5] Figure 3 shows the p-diagram fragment for the six syllables of *der Partner der Freundin*. The 'raw' signal information given in Figure 3 encodes patterns which can be interpreted in categorical terms at the *interpretation level*. For example, a strong rise in F_0, a following drop (from S_2 to S_4) and a comparatively long duration on the last (unstressed) syllable of *Partner* (as seen at S_3: [n6]) are strong indicators for a phonological phrase boundary. As a result, PHRASING $=)_\varphi$ is added to the syllable's vector at the interpretation level (Figure 4). Further possibilities at the interpretation level include, for instance,

...	...	...	...	...	...		interpretation
PHRASING	-	-	$)_\varphi$	$(_\varphi$	-	-	↓
SEMIT_DIFF	...	-1	6.8	-4.3	-1.9	2.6	
GToBI	-	L*	+H H-	-	L*	+H	
DURATION	0.15	0.25	0.25	0.13	0.31	0.19	signal
FUND. FREQ.	192	181	269	209	188	218	↓
SEGMENTS	[de:6]	[pa6t]	[n6]	[de:6]	[fROYn]	[dIn]	
VECTORINDEX	**S_1**	**S_2**	**S_3**	**S_4**	**S_5**	**S_6**	

Figure 4: The interpretation level of the p-diagram for *der Partner der Freundin*

[4] Segments are represented in SAMPA, a computer-readable phonetic alphabet (Wells 1997).

[5] Mean F_0 is calculated based on the complete syllable and serves as a quick orientation for the researcher, not as a basis for the computational calculation discussed below.

a GToBI (Grice and Baumann 2002) analysis of the pitch in terms of high and low tones, or the differences between adjacent pitch values measured in normalized semitones (SEMIT_DIFF), which allow for an interpretation of the slopes leading to and from the accent (i.e., the scaling of the tones). While the p-diagram representation was developed with LFG in mind, it is an encapsulated, adaptable, and extendable representation that can be plugged into any modular framework.

The transfer of vocabulary 3.2

The transfer of vocabulary associates morphosyntactic and phonological information in lexical elements via the multidimensional lexicon. Following proposals made by, e.g., Levelt *et al.* (1999), the lexicon includes several dimensions (Table 3): The *s(yntactic)-form* contains the traditional morphosyntactic information associated with a particular lexical item (e.g., number, gender, or case), while the *p(honological)-form* contains information on the segments and the metrical frame of that entry: the number of syllables, the lexical stress pattern, and the prosodic status (e.g., whether the element is a clitic, underspecified, or a prosodic word). The lexicon in Table 3 shows the entries for the noun *Freundin*, which is feminine, singular, and a prosodic word with two syllables in a trochaic foot structure. The determiner *der* has ambiguous case information (genitive or dative) and consists of a single, prosodically underspecified syllable.[6] The lexicon is modular in that

s-form				p-form	
N	(↑ PRED)	=	'Freundin'	SEGMENTS	/f R OY n d I n/
	(↑ NUM)	=	sg	METRICAL FRM	$(\text{'}\sigma\sigma)_\omega$
	(↑ GEND)	=	fem		
D	(↑ PRED)	=	'der'	SEGMENTS	/d e 6/
	(↑ NUM)	=	sg	METRICAL FRM	σ
	(↑ GEND)	=	fem		
	(↑ CASE)	=	{gen | dat}		

Table 3: (Simplified) lexical entries for *der* and *Freundin*

[6] The determiner 'der' can also be used in the nominative masculine. This option is omitted from Table 3 since it is not relevant for the data discussed in this paper.

there is a strict separation of module-related information: Each lexical dimension can only be accessed by the related module, i.e., p-structure can only access p-forms, and c-structure can only access s-forms. At the same time, the lexicon has a translating function: Once a dimension is triggered, the related dimensions can be accessed as well. During comprehension, if p-structure accesses a particular p-form, the related s-form becomes available and the morphosyntactic information is instantiated to syntactic structure. Conversely, during production, if c-structure accesses an s-form, the related p-form information becomes available to p-structure, ultimately forming the foundation for the phonetic utterance.

3.3 *The transfer of structure*

The transfer of structure exchanges information on prosodic and syntactic constituency via the projection function $\natural$. Figure 5 shows the annotation for an object nominal phrase (NP) which checks whether there is a (left) phonological phrase boundary associated with the left edge of the NP's corresponding prosodic unit in p-structure. The annotation can be read as follows: For all terminal nodes T (= {D/*der*, N/*Freundin*}) of the current node * (=NP), for the syllable with the smallest index ($S_{\min}$) in this set of terminal nodes (i.e., the leftmost syllable), there must be ($=_c$) a (left) phonological phrase boundary $(_\varphi$. If this is the case, an object with dative case is projected to f-structure: (↑ OBJ) = ↓ and (↓ CASE) = dat state that any material occurring under the current syntactic node (here: NP) is stored as part of the

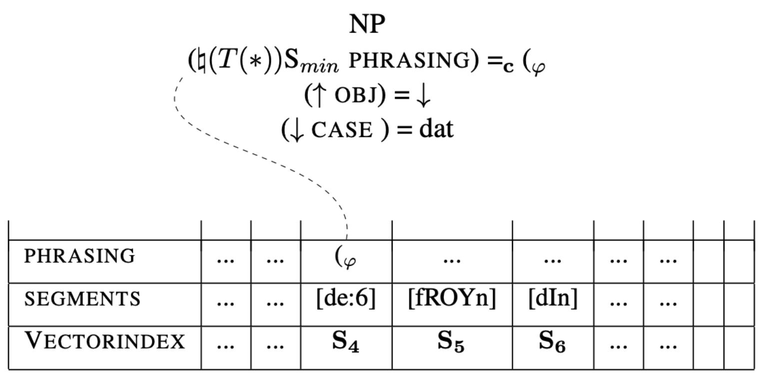


Figure 5: The *transfer of structure*: prosodic and syntactic phrasing

grammatical function 'object' in f-structure, and that the related case is dative.[7] The annotation of the c-structure node NP thus combines two projection functions: First, the information concerning prosodic phrasing at p-structure is determined. If a prosodic phrase boundary is present, the current node is then interpreted as the object of the clause, effectively disambiguating the syntactically ambiguous structure in example (3)/Figure 1.

Figure 6 shows the complete analysis of a dative structure at the prosody-syntax interface during comprehension, where the transfer of

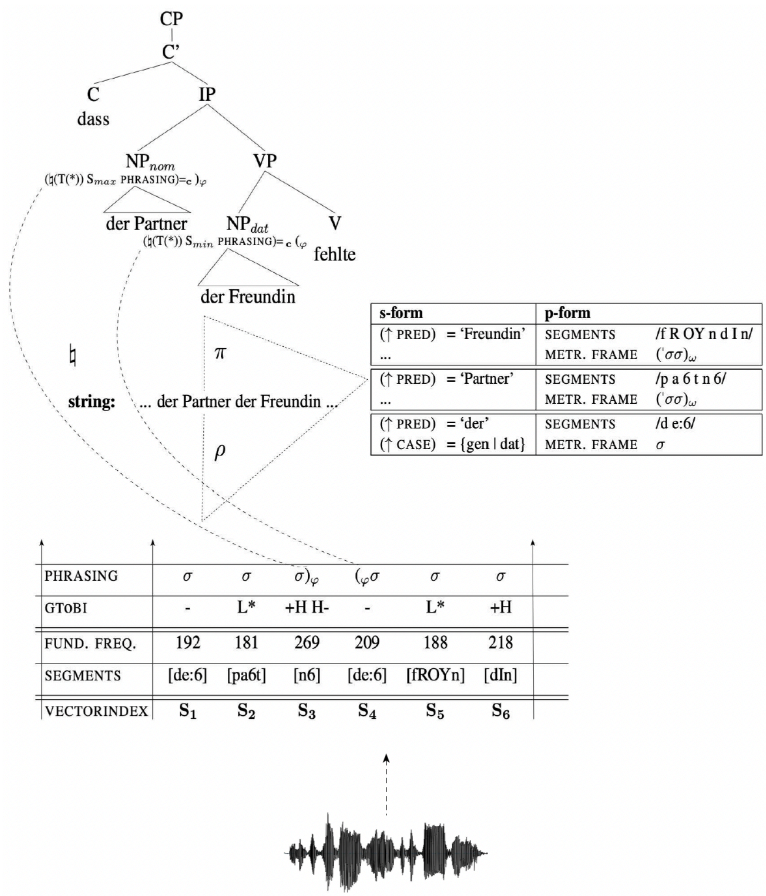


Figure 6: A dative structure at the prosody-syntax interface: comprehension

[7] For further explanations of the correspondence between c- and f-structure, the interested reader is referred to Dalrymple 2023.

vocabulary matches segmental strings against lexical items and the transfer of structure disambiguates the syntactically ambiguous structures based on larger prosodic constituents, in this case a phonological phrase boundary between the two NPs [*der Partner*] and [*der Freundin*].

This section provided the theoretical background for the prosodic disambiguation of syntactically ambiguous structures in LFG. The following section takes this theoretical analysis as a starting point and serves as a blueprint for an integration of prosodic structure into the existing computational LFG-grammars, thus enabling the grammars to include and process information from the speech signal as well.

4 COMPUTATIONAL IMPLEMENTATION

The computational implementation of the theoretical analysis presented in Section 3 is a new approach that includes the integration of spoken language. It categorizes the gradient information gained from the signal and organizes it within the p-diagram at p-structure. It then matches the information against a lexicon containing p-form and s-form information. The matching process leads to the creation of the s-string which is the linear concatenation of all matched s-forms and thus corresponds to the string that was originally used as input to the computational LFG grammars. The s-string (and the lexical morphosyntactic information associated with each word in the string) enables c- and f-structure to be parsed with XLE (Crouch *et al.* 2017), the grammar development platform used to create large-scale LFG grammars. In a final step, the implementation allows for the disambiguation of syntactic structures based on the automatically determined prosodic phrase boundaries at p-structure. The implementation is in Perl, with added scripts written in Praat (Boersma and Weenink 2021), xfst (Beesley and Karttunen 2003) and R (R Core Team 2016), all of which are open-source and commonly used software.[8]

[8] The source code for the computational implementation is available under https://github.com/ticle2/prosody-syntax-interface-in-LFG.

Extracting and normalizing information from the speech signal 4.1

Figure 7 shows the input used for the computational implementation, a sound file annotated with SAMPA syllables. For the annotation, the data was first automatically annotated using the Munich Automatic Annotation System MAUS (Kisler *et al.* 2017; Schiel 1999), which aligns the speech signal with SAMPA segments (but not syllables) based on a given orthographic input. In order to obtain the syllabic annotation that serves as a base for the system described in this paper, the segmental annotation was matched against a lexicon created from the CELEX database for German words (Baayen *et al.* 1995). This database allows for the creation of different custom-tailored lexicons, in this case a lexicon containing the SAMPA-syllables for all the German words in the database. In a next step, the segmental MAUS annotation was matched against the syllable-based lexicon, keeping track of the start and end times of each syllable in the speech signal. Based on this information, a new Praat annotation tier was created containing only the SAMPA syllables. The syllable tier was then manually checked for alignment mistakes that regularly occur with forced

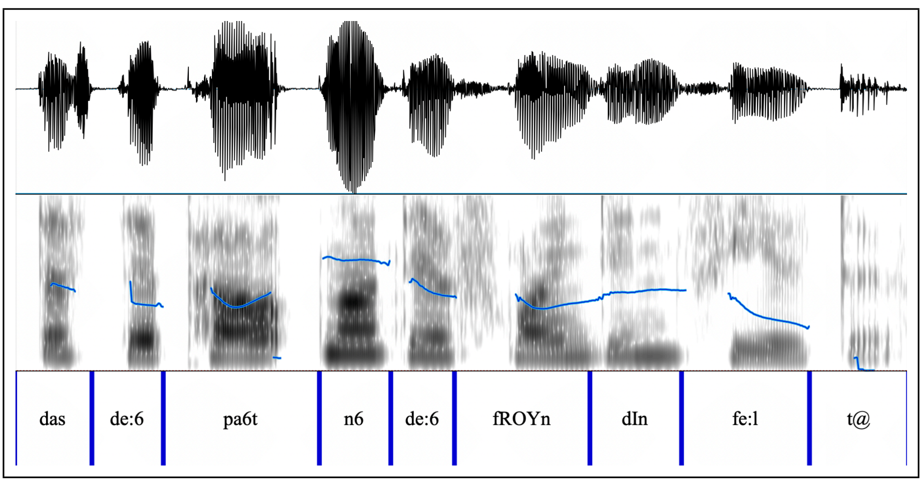


Figure 7: Input: a sound file annotated with SAMPA syllables in Praat, here for example (3)

aligners like MAUS (see, e.g., Gonzalez *et al.* 2020).[9] In a first step, a Praat script collects information from the speech signal. The script extracts the syllable segments, the duration of each syllable, and the mean F_0-values for each syllable for the signal level of the p-diagram (Figure 3). For a fine-grained analysis of the pitch during processing, the script furthermore divides each syllable into five even-spaced subintervals, takes the mean F_0-values of each subinterval and turns these values into semitones, thus effectively normalizing duration and pitch. In order to minimize the effect caused by incorrect pitch calculations by the Praat algorithm, the system checks for outliers among the semitones and – if present – excludes them from the following estimation of high and low tones.[10]

Each subinterval is tagged for position within the syllable, either as central, or as preceding or following a syllable boundary. This measure was implemented to allow for the determination of early or late pitch accents. For example, if a pitch accent unexpectedly occurs in an unstressed syllable preceding the stressed target syllable, the information that it occurs directly at the boundary to the target syllable would relate this accent to the target syllable as an 'early' accent.

4.2 *Interpreting the pitch*

In a second step, the raw values from the speech signal are interpreted in terms of categories that are 'meaningful' for other modules of grammar. Different measures are used for the interpretation of the pitch: In addition to the semitones and the differences between these semitones indicating falls and rises, the implementation also utilizes the residuals of a linear regression based on the pitch values of a given speech

[9] It would, of course, be desirable to have a system that provides a deep linguistic analysis from the raw speech signal to a syntactic structure. However, the fact that forced alignment of orthographic text to segmental annotation requires manual correction by a human annotator means that uncontrolled alignment (i.e., without the orthographic representation) would most likely result in increased inaccuracy. Since the main focus of this paper is on the implementation of the prosody-syntax interface, and not automatic speech recognition, the system starts with input files that are annotated with SAMPA syllables.

[10] Where an outlier is any data point above the 3rd quartile +1.5 Interquartile range (IQR) and below the 1st quartile −1.5 IQR (e.g., Winter 2019, 60).

signal. This measure was introduced to account for the lowering or rising of the pitch over time depending on the sentence type; e.g., in declaratives the pitch tends to get lower towards the end of the sentence (a.o., Ladd 1984; Xu 2005). This general tendency is reflected by the regression line. Residuals return the distance of each value from this line and are thus a good measure to describe deviations from the average, i.e., surprising values.

Both semitones (and their differences) and residual values are then used a) to determine the minimums (L) and maximums (H) in a given signal, and b) to determine the slopes between these categories, i.e., whether the rises/falls are strong or weak.

In order to mark both categories (i.e., type of accent and type of slope) in one representation, we devised the system in Table 4: Each level of L or H is characterized by a particular height and shape of the slope leading to it (*lead*) and following it (*tail*). Taken together,

Cat.	Max/Min	Lead	Tail
H4/L4	Max/Min	steep	steep
H3/L3	Max/Min	steep	flat
H2/L2	Max/Min	flat	steep
H1/L1	Max/Min	normal	flat

Table 4:
(Part of the) system of pitch accents and slopes in the computational implementation

semitones and residuals allow for the detection of deviations from the norm in the signal, i.e., maximums (H) and minimums (L). In order to exclude microprosodic effects (which might cause two tones to appear on one syllable), the distance between any Hs and Ls has to span at least one syllable. Slopes to and from a H/L tone are calculated based on the ratio between the semitones of adjacent Ls and Hs and the distance (the number of subintervals) between them. The resulting values indicate whether the associated slopes are steep or flat.H4 and L4 thus represent accents where the lead and the tail show a strong rise and fall respectively, while H1 and L1 have a relatively flat lead and tail. L2/L3 and H2/H3 are positioned between these two extremes, with each having a slightly different shape depending on the slopes. The following Figure 8 demonstrates some of the H and L tones discussed in Table 4 for a dative example as they would be assigned by the system. The tone values are stored as part of the *interpretation* level

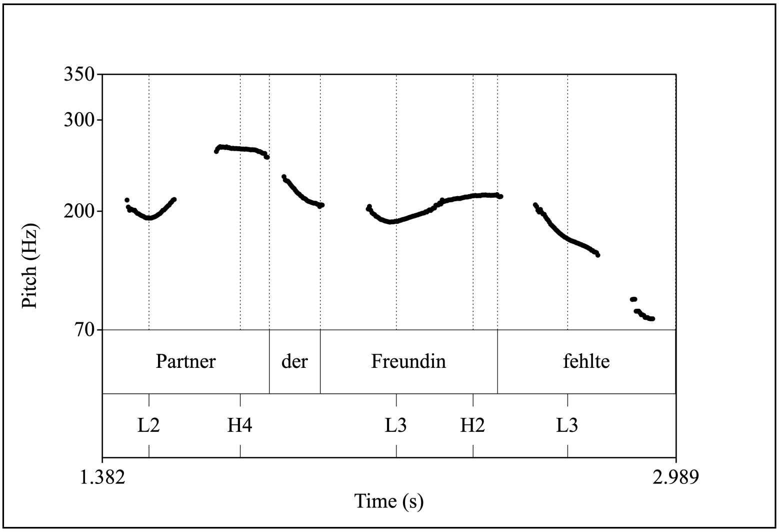


Figure 8: Tone values for a dative example as calculated by the system

of the p-diagram (Figure 4), where they replace the traditional GToBI values in order to facilitate (and simplify) the automatic interpretation by other modules of the grammar.

4.3 *Interpreting duration*

The categorization of a specific syllable as 'long' or 'short' is not a trivial process. Since the input to the system is always a single file, there is no direct way to compare the duration of one syllable to duration measures of syllables in similar positions in other input signals.[11] For the current analysis, this problem was resolved by creating a precompiled threshold for duration categorization. The compilation was based on the 270 utterances produced in the experiment described in

[11] There are two ways to deal with this problem: a) a database of all possible syllables in all possible (word) positions over many speakers in order to get an estimate of the expected syllable duration, or b) an estimation of syllable duration tailored to the dataset at hand. While a database would allow a more universal assessment of syllable duration, creating such a resource would be very time-consuming and the considerable size of such a database would be more of a hindrance to the system at hand. Since this paper is a proof of concept, we leave this work to further research, and show how option b), a tailored solution, can be realized.

Section 2, more precisely, on the stretch of data from the start of the subordinate clause to the end of the second noun; 7 syllables in total. Strictly speaking, the verb should have also been analyzed as part of this clausal stretch. However, it was disregarded for this particular calculation because different verbs show too much segmental variation. This, in turn, would have had an (undesired) effect on the duration measures.

Two values were used to classify syllables as long or short: speaker tempo and syllable duration. For the estimation of speaker tempo, the duration of each of the seven target syllables was added up for each single recording and then divided by 7. The resulting values for each signal produced by a single speaker were added up again and the mean over all values was calculated. This mean value was taken to represent the individual speech tempo for each speaker. The following Table 5 shows the distribution of speaker tempo values over all speakers. As we can see, the 'fastest' speaker has a rate of 0.150 seconds per syllable and the 'slowest' speaker has a rate of 0.225. The overall mean was 0.184. For the categorization into slow and fast speakers, the first and third quartile (0.170 and 0.196 respectively) were used as thresholds. Values below/above these thresholds can be deemed unexpected from a statistical perspective, so any speaker with a value below 0.170 could safely be considered as 'fast', and any speaker above 0.196 as 'slow'.

Minimum	1st Quartile	Mean	3rd Quartile	Maximum
0.150	0.170	0.184	0.196	0.225

Table 5:
Distribution of speaker tempo values over all speakers in seconds per syllable

In addition to speaker tempo, we also determined the duration of each individual syllable in the target area in comparison to all syllables in the same position in the overall dataset, e.g., each first syllable in the first noun was compared to all other syllables that also occurred in the first position of the first noun. For these values, the mean duration of each syllable over all the speakers was taken; outliers were excluded.

For the fourth syllable in the target area (which corresponds to the second syllable of the first noun, e.g. [nɐ] in *partner*), we observed the distribution in Table 6. As discussed in Section 2, the fourth syllable

Table 6: Distribution of duration values for the fourth syllable in the target area over all speakers in seconds per syllable

Minimum	1st Quartile	Mean	3rd Quartile	Maximum
0.1579	0.1681	0.1783	0.1885	0.1987

is significantly longer in the dative condition than in the genitive, thus signalling a prosodic phrase boundary. Syllables with a duration above the 3rd quartile were interpreted as 'long' (= increased likelihood of boundary), and syllables below the 1st quartile as 'short' (= no boundary following).

While this estimation of expected and unexpected values of syllable duration is a good indication of a following prosodic phrase boundary, any duration value needs to be viewed with reference to speaker tempo. The reason is that a slow speaker will per se also produce slow syllables which will confound the calculation of a prosodic phrase boundary. To control for this particular factor, syllables were only categorized as slow if the speaker had a fast or normal speaking rate. For slow speakers producing slow syllables, the difference between the speaker's tempo and the overall mean speaker tempo was taken and subtracted from the duration value of the syllable in question. If this syllable could still be classified as slow, the value was retained.

Both speaker tempo rates and individual syllable duration are stored as part of the system and are accessed during signal interpretation in order to facilitate boundary calculation.

4.4 *Lexical matching: the transfer of vocabulary*

During the transfer of vocabulary, the input from the speech signal is matched against the p-forms of the multidimensional lexicon, which then makes the associated s-forms available for syntactic parsing. In order to acquire the correct s-string, the p(honological)-string, which is created by concatenating the SAMPA syllables from the input speech signal (… *de:6.pa6t.n6.de:6.fROYN.dIn* …), is matched exhaustively against a lexicon including phonological and morphosyntactic material as described in Section 3, Table 3. The lexicon is a finite-state

Input (p-string)	**Lexicon**		**Output (s-string).**
	p-form	**s-form**	
... de6.fROYn.dIn ... →	de:6	der	→ ... der Freundin ...
	fROYn.dIn	Freundin	
	...	...	

Table 7: The transfer of vocabulary: from p-string to s-string

transducer (xfst; Beesley and Karttunen 2003), where the upper side corresponds to the s-form, and the lower side to the p-form information associated with the lexical item. Matching the p-string against the lexicon results in the corresponding s-string (... *dass der Partner der Freundin* ...), which constitutes the input for the syntactic structure. Apart from making the s-string and the associated morphosyntactic information available to c- and f-structure, the matching of the p-string against the lexicon also makes the lexical p-form information (e.g., information on lexical stress or prosodic word/clitic status) available to the p-diagram.[12]

4.5 *Prosodic phrase boundaries and the p-diagram*

The previous sections described the different aspects relevant for the representation of a speech signal at p-structure. As a last step, prosodic phrase boundaries are calculated.

The production experiment reported in Section 2 elicited the acoustic factors which can be relevant for the determination of a PhP boundary: a rise in F_0 towards the boundary followed by a drop after the boundary, a pause, and a relatively long pre-boundary syllable.

[12] This information is especially relevant for production (not discussed here), because it allows the modelling of a prosodic baseline that can later be 'translated' into phonetic terms. But it is relevant for comprehension as well, in that it is generally assumed that pitch accents are only associated with lexically stressed syllables in German. Due to vowel quality differences and other reasons, however, the algorithm might also determine the local maximum or minimum to be on the previous or following syllable. Lexical stress (possibly in combination with positional information of the accent in the syllable, cf. Section 4.1) could in principle be used to shift the accents to the target syllable in the p-diagram representation.

Based on the pitch calculations in Section 4.2, the duration values in Section 4.3, and on the presence or absence of pauses, the implementation estimates the likelihood of a prosodic phrase boundary in the position at hand. If any of the following constraints are minimally met, a PhP boundary is included.

1. a H4 accent
2. a H3 accent in combination with a surprising residual value; only very high values (above 3 or below −3) are taken into account
3. a H3 accent with a long syllable
4. a pause

Figure 9 shows an automatically created p-diagram for the string *der Partner der Freundin* based on a speech signal with the dative construction. As discussed in Section 3.1, each vector includes the segments, the duration, and the mean F_0-value for the associated syllable. The p-diagram also contains the lexical p-form information by marking lexically stressed syllables with x and by adding the lexical prosodic unit information to the attribute PROS_PHRASE (prosodic phrasing). While each function word (*dass, der*) is represented by an underspecified syllable σ, the nouns' prosodic word status is indicated by a set of unmarked brackets. The automatically calculated PhP boundaries are marked by $_{pp}($ and $)_{pp}$. The p-diagram in Figure 9 shows that the system can give a fairly accurate categorical representation of the speech signal. The PhP boundary occurs after the first NP, thus indicating a dative structure. There are also several open questions, e.g., whether the low tone L2 associated with vector 2 (GToBI: L*), which occurs just before the syllable boundary, should be 'moved' to vector 3 where the syllable carries lexical stress, or whether an additional attribute for

pros_phrase	pp(σ	σ	(σ	σ))pp	pp(σ	(σ	σ)	(σ	σ))pp
pitch_tones		L2		H4		L3	H2	L3	
lex_stress	–	–	x	–	–	x	–	x	–
F0_mean	225.62	196.49	198.90	267.53	219.35	194.02	213.77	176.27	85.71
duration	0.17	0.16	0.33	0.18	0.14	0.30	0.20	0.28	0.22
syllables	das	de:6	part	n6	de:6	frOyn	dIn	fel	t@
Vector_index	1	2	3	4	5	6	7	8	9

Figure 9: P-diagram for a dative interpretation of the string *der Partner der Freundin* ('the partner of the friend')

'early' or 'late' L/H tones would be more useful. We leave questions like these to further research.

The information on prosodic phrase boundaries at p-structure is now available for further processing. However, in order to disambiguate the syntactic structure, c-structure has to recognize the ambiguity in the first place and be able to check for possible cues for a particular interpretation at p-structure.

Disambiguation and the fchart: the transfer of structure 4.6

This section describes how the overgeneration caused by syntactic ambiguities as in example (3) can be automatically disambiguated by intersecting a computational LFG-grammar for German with the p-structure created above.[13] In a first step, the syntactic string determined in Section 4.4 is parsed with a computational LFG grammar. In order to achieve this, the main Perl script creates an XLE-internal *xlerc* script (Crouch *et al.* 2017) which starts the computational grammar and parses the s-string. As expected, the grammar overgenerates and returns the two syntactic strings in Figure 1. The syntactic ambiguity leading to these parses can be made accessible by instructing the xlerc script to print out the so-called *fchart*, a Prolog representation of all choices, constraints, c-structure relations and more, in one file. The command in (4) will return a Prolog file *filename.pl*, which can be processed further by the main Perl script.

(4) `print-prolog-chart-graph filename.pl`

The following descriptions discuss only the relevant parts of the extensive fchart Prolog representation and the way they can be used to determine the actual linear position of the ambiguity (with the ultimate goal to check for prosodic phrase boundaries at that position).

The fact that there are two possible syntactic structures is encoded in the fchart section 'Choices' with the variables A1 and A2 (in this example, A1 corresponds to the dative option, and A2 to the genitive). This information alerts the script to the ambiguity of the parsed syntactic string.

[13] The following discussion describes this process in some technical detail; readers who are not familiar with XLE might want to continue with Section 5.

(5)
```
[
choice([A1,A2], 1)
],
```

The next fchart section ‘Constraints’ indicates that the two choices A1 and A2 are based on the ambiguity in the verb’s valency.

(6)
```
% Constraints:
[
cf(A1,eq(var(3),semform('fehlen',4,[var(4),var(2)],[]))),
cf(A2,eq(var(3),semform('fehlen',4,[var(4)],[]))),
],
```

As shown in (6), the verb *fehlen* ‘miss’ in choice A1 has two arguments (represented by abstract variables, var(4) and var(2)) and in choice A2 only one argument (var(4)). With respect to the linguistic example discussed in this paper, choice A1 thus refers to the (transitive) dative, i.e., to the two arguments [*der Partner*] and [*der Freundin*], and choice A2 to the (intransitive) genitive with one nested argument [*der Partner* [*der Freundin*]]. This difference in argument structure and the variable names of the arguments for each choice are then further tracked by the main script in order to ultimately relate these abstract variables to concrete surface forms.

In the fchart section ‘C-structure’ in (7), the *fspans* of the arguments (i.e., the s-forms over which the argument ‘spans’) are encoded with indexing numbers, where the first number indicates the start of the span, and the second number the end of the span. In example (7), the two arguments in choice A1 have the fspan from 17 to 28 for the first argument var(4) and the fspan from 29 to 41 for the second argument var(2). For choice A2, the single argument var(4) has an fspan from 17 to 41 (notably including the range of both arguments in choice A1).

(7)
```
% C-Structure:
[
...
cf(A1,fspan(var(4),17,28)),
cf(A1,fspan(var(2),29,41)),
...
cf(A2,fspan(var(4),17,41)),
...
```

These numbers correspond to the surface forms (i.e., the s-forms or terminal nodes at c-structure). They indicate the start and the end of each of the arguments. In the next step, the script relates these index numbers from the fspans to the surface forms. Index number 17, the starting position of the first argument var(4) in both option A1 and A2 (cf. (7)), is associated with the start of the (first) determiner *der* of the first NP [*der Partner*] shown in the fchart excerpt in (8).

(8) `cf(1,surfaceform(9,`**'der',17,20**`))`
→ start of the first argument var(4) in both options

In choice A1, the span of the first argument var(4) is terminated with the indexing number 28, which also indicates the end of the surface form *Partner* in example (9). The first argument var(4) in choice A1 (but not A2) is thus the NP [*der Partner*].

(9) `cf(1,surfaceform(11,`**'Partner',21,28**`))`
→ end of the first argument var(4) in option A1 (subject in the dative construction)

As shown in (10), the surface form of the determiner of the second NP starts with index number 29. As seen in (7), this is also the start of the second argument var(2) in choice A1.

(10) `cf(1,surfaceform(13,`**'der',29,32**`))`
→ start of second argument var(2) in option A1

Finally, the surface form *Freundin* ends with index number 41, the terminating index number of the second argument (var(2)) of choice A1, and of the first and only argument (var(4)) of choice A2.

(11) `cf(1,surfaceform(15,`**'Freundin',33,41**`))`
→ end of second argument var(2) in option A1 (object of the dative)
→ end of first argument var(4) in option A2 (subject of the genitive)

For choice A1, the second argument thus stretches from the beginning of the second determiner (index 29, (10)) to the end of *Freundin* (index 41, (11)): [*der Freundin*]. In contrast, the argument for choice

A2 stretches all the way from the beginning of the first determiner (index 17, (8)) to the end of *Freundin*: [*der Partner der Freundin*].

By going through the fchart step by step, following each of its two choices A1 and A2, the script can pinpoint the position of the ambiguity in the syntactic string. In this case, this is the position at the end of the first NP [*der Partner*], where choice A1 concludes the first argument, and choice A2 does not.

Since the edges of syntactic NPs are associated with PhP boundaries (as established with the production experiment in Section 2), the algorithm now needs to check whether there is a PhP boundary after the last syllable of *Partner* in the p-diagram created in the last section. If this is the case, then choice A1 (the dative) should be selected, because we would expect a PhP boundary to be present between the two arguments. If there is no PhP boundary then choice A2 (the genitive) is more likely because the single argument should not be 'interrupted' by a PhP boundary. The selected option can be encoded in the Prolog file by automatically rewriting the fchart section 'Choices' which originally contained both choices (see (5)). The following example shows the 'Choices' section rewritten for choice A1 (i.e., the dative).

(12)
```
[
select(A1, 1)
   ]
```

In the last step, the main script starts an xlerc script containing the command in (13) which reparses the altered fchart. Since only one choice (A1) is given, the script only pays heed to the structures and constraints associated with that choice and ignores the others, thus effectively disambiguating syntactic structure by means of prosodic information.

(13) `read-prolog-chart-graph filename_new.pl`

Figure 10 shows XLE's c-structure output after the script reparsed the disambiguated fchart.

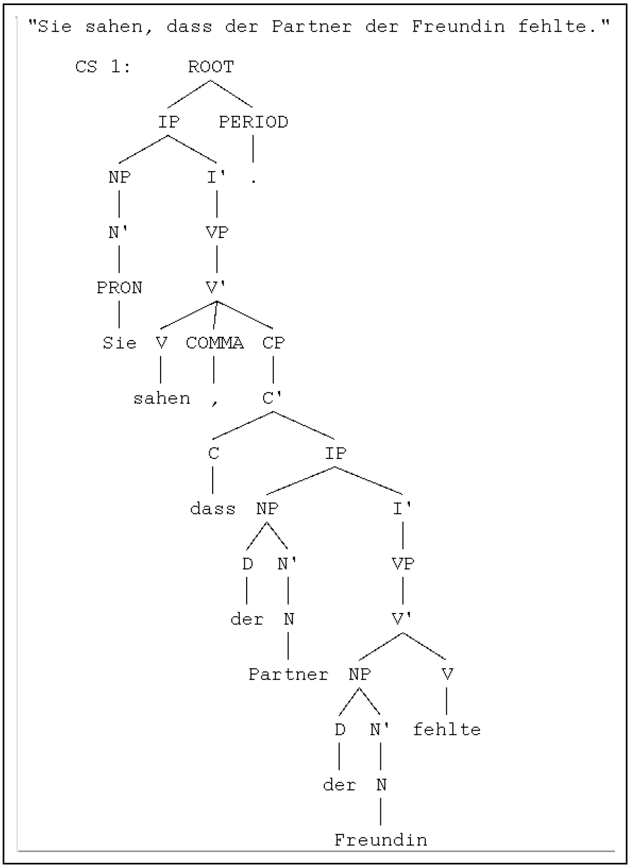


Figure 10: A prosodically disambiguated dative c-structure

5 EVALUATION

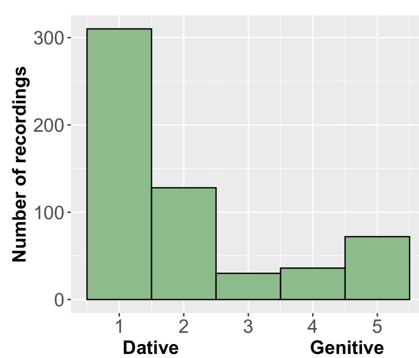
For the evaluation, the recordings described in Section 2 were used to create a gold standard. Since spoken data has a lot of variation (with statistical analyses mostly only capturing tendencies), the data first needed to be sorted into representative and non-representative recordings for each case condition. To this end, a perception study was conducted in order to determine which of the recordings were most likely to be interpreted as datives or genitives by listeners. The motivation for this approach is to only evaluate the system on recordings that human listeners would be able to identify as well.

5.1 *Perception study*

In an online perception experiment, 32 native German speakers were asked to rate the 270 recordings from the production study described in Section 2. For the experiment, the recordings were randomized and assigned to different experimental lists. Each participant was asked to listen to nine genitive and nine dative recordings, and to indicate which meaning they thought was associated with the signal on a scale from one to five. On the scale, 1 (and to a lesser degree 2) represented a dative interpretation, 3 was considered ‘undecidable’, i.e., listeners did not show a clear tendency towards the case condition, and 5 (and to a lesser degree 4) represented the genitive interpretation. Each sentence was rated by two or three listeners (depending on the list), resulting in a total of 576 ratings. Only the sentences that were correctly rated at least twice (i.e., where the case of the produced sentence matched the case perceived by the listeners) were included in the gold standard and used for the evaluation.

Although datives and genitives were evenly distributed in the presented material, listeners were much more likely to mark a recording as dative. Figure 11 shows the distribution of listener responses over all recordings. A non-parametric two-tailed Wilcoxon rank sum test showed that the response values differed significantly from the actual case values ($W = 35765$, $p < 0.01$). Table 8 shows the 576 rating responses of the perception experiment where 32 listeners each rated 18 (9 dative and 9 genitive) randomized recordings. The results confirm that the mismatches between listener responses and actual case values were particularly high for the genitive recordings. This

Figure 11:
Listener ratings of (equally distributed) dative and genitive recordings



	Dative	Genitive
Match	238	74
Mismatch	50	214

Table 8:
Matching and mismatching occurrences between listener ratings and actual case condition

mismatch is likely to be due to the general historic decline of the genitive in comparison to the dative (see, e.g., Scott 2011; Pittner 2014). As a consequence, the recordings that made up the gold standard were imbalanced between the two case conditions: From the original 270 recordings, 78 were categorized by at least two listeners as dative (1 or 2) and 17 as genitive (4 or 5). Note also that the recordings of one of the 15 speakers that took part in the production experiment described in Section 2 never received correct ratings by the participants of the perception study, i.e., this speaker did not use the prosodic cues that were necessary for the listeners to disambiguate the syntactic structure. For this reason, the following evaluation is only based on the recordings from 14 speakers.

5.2 *Evaluation*

In a next step, the gold standard recordings were semi-automatically annotated with SAMPA syllables following the process described in Section 4.1. Input to the system was a single wav-file with a corresponding TextGrid containing one Tier with SAMPA syllables (as illustrated in Figure 7). Each output by the system was checked for syntactic disambiguation and the placement of a correct prosodic phrase boundary in the target position in the p-diagram.

We present two types of evaluations. The ‘broad’ evaluation includes ratings that only show a tendency towards a particular interpretation: If two listeners rated a recording with a 2 or if there were mixed ratings (1 and 2), the recording was still classified as a dative even though the choice of rating showed some insecurity. By contrast, the ‘narrow’ evaluation only included recordings where all listeners were confident of the interpretation, i.e., all of them uniformly rated a dative with a 1 and a genitive with a 5. The reason for this distinction will become clear below.

5.2.1 Broad evaluation: results

The broad evaluation included 78 dative and 17 genitive recordings. The system was able to correctly interpret 68 of the 95 cases (71.5%). Figure 12 shows the results for each case condition and for both conditions taken together. Table 9 shows the system's performance measures for the broad evaluation. Since the data used for the broad evaluation still contained a level of insecurity (ratings 2 for a dative and 4 for a genitive), the evaluation was repeated including only the recordings where at least two speakers unanimously rated a dative as 1 or a genitive as 5.

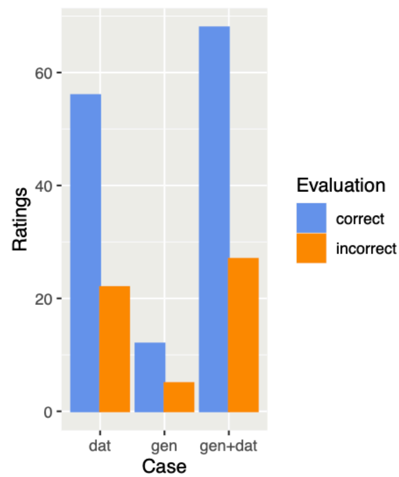


Figure 12: Correctly and incorrectly labelled input signals sorted by case condition; broad evaluation

Table 9: Precision, Recall, and F_1-score measures for the broad evaluation

	Precision	Recall	F_1-score
Dative	0.918	0.718	0.806
Genitive	0.353	0.706	0.471
Macro-average	0.636	0.714	0.639

5.2.2 Narrow evaluation: results

For the narrow evaluation, only recordings rated confidently as dative or genitive (i.e., 1 or 5) by at least two listeners were used. This resulted in 48 recordings for evaluation (38 dative, 10 genitive). The system was able to correctly determine 79% of the input (see Figure 13),

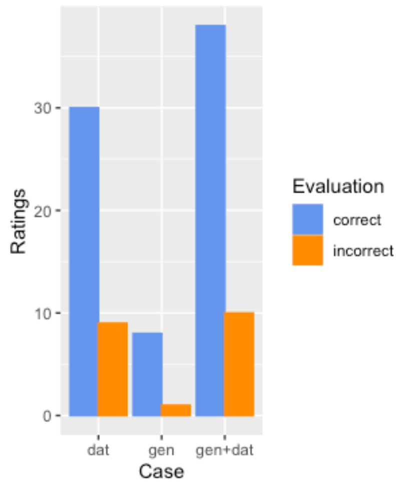


Figure 13:
Correctly and incorrectly labelled input signals sorted by case condition; narrow evaluation

	Precision	Recall	F_1-score
Dative	0.968	0.769	0.857
Genitive	0.471	0.889	0.616
Macro-average	0.72	0.829	0.737

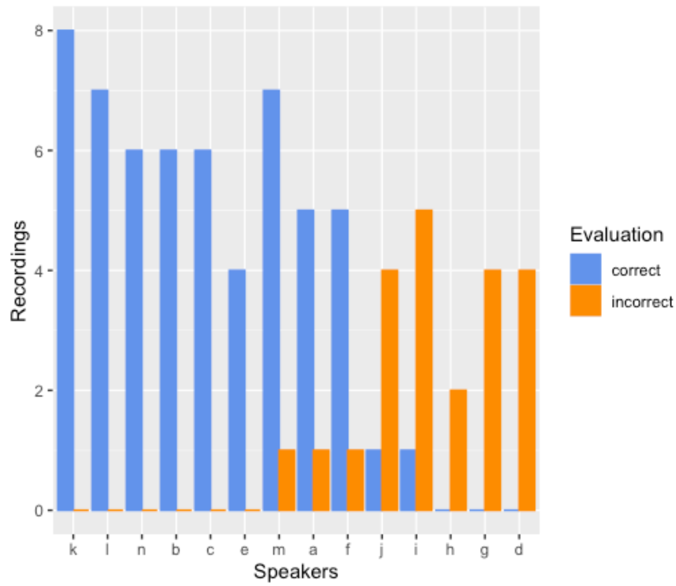
Table 10:
Precision, Recall, and F_1-score measures for the narrow evaluation

i.e., the results are noticeably higher compared to the broad evaluation where tendencies (2 and 4) were included as well. The better performance of the system in the narrow evaluation is also reflected in the performance measures in Table 10. Although these values are promising, there are still a number of recordings which were correctly identified by the listeners, but not by the system. The following section discusses some additional findings and possible reasons for this difference.

5.3 *Discussion*

As discussed above in Section 5.1, the evaluation data is based on 14 out of 15 speakers who took part in the original production study, as none of the recordings by speaker no. 15 were correctly rated by the participants of the perception study. Furthermore, the data is very imbalanced, which reflects a more general preference by speakers to use the dative instead of the genitive in object constructions. However,

Figure 14:
Results of the broad evaluation of the dative recordings sorted by speaker and correctness of the evaluation



as the system is not trained on corpus data, it is not affected by this preference for the dative construction.

A closer look at the data reveals strong speaker variation as illustrated in Figure 14, where the correct and incorrect evaluations for the dative are displayed for each speaker separately. According to Figure 14, the speakers can be sorted into two categories: Speakers whose recordings were rated correctly by both the system and the listeners (blue), and those whose recordings were rated correctly by the listeners, but not the system (orange). While there are five speakers with both correct and incorrect system ratings, the speakers can still be clearly associated with one of the two groups. In experimental terms this means that there is a group of speakers who (predominantly) signal the dative by acoustic means which were not captured by the production experiment described in Section 2; i.e., these speakers do not use pitch movement, a pause, or duration as acoustic cues at the target position between the two NPs. As a consequence, the system cannot correctly distinguish between the two syntactic structures.

Since the experiment was designed specifically for this target position, it is at this point not possible to determine the strategy used by this subgroup (this has to be left for future research).

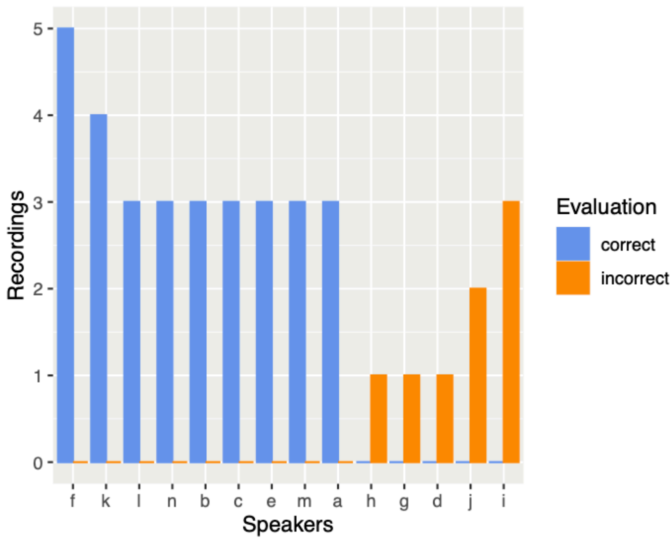


Figure 15: Results of the narrow evaluation of the dative recordings sorted by speaker and correctness of the evaluation

As with the broad evaluation, speaker variation was clearly visible in the narrow evaluation as well: Figure 15 shows a precise division between two groups of speakers.[14] The subgroup {j, i, h, g, d} does not seem to use the acoustic cues to signal a prosodic boundary discussed in Section 2 and can thus not be correctly classified by the system.

6 CONCLUSION

This paper introduced a new end-to-end system, which takes a speech signal annotated with syllables as input, extracts the different acoustic cues, calculates pitch accents and prosodic phrase boundaries, and creates a representation of the data in form of a p-diagram. The information stored in the p-diagram is subsequently used by a computational LFG grammar to disambiguate the syntactically ambiguous structures in the input.

The implementation enables the traditionally text-based computational LFG grammars to process spoken language and to integrate

[14] One has to keep in mind that the data itself is greatly reduced here, with speakers h, g, d only contributing a single sentence.

the speech signal information into the analysis of linguistic phenomena, thus closing the gap between automatic speech recognition and linguistically deep computational grammars. In addition to syntactic and semantic analyses, the computational LFG grammars can now process and interpret any phenomena indicated by prosodic constituency or pitch accents. As such, they take a major step towards real automatic speech understanding.

An initial evaluation of the German system showed promising results and gave interesting insights into speaker variation. Challenges are manifold, and foremost is the problem that prosody is always gradient and includes a lot of variation (within and between speakers, but also within and between different language varieties, etc.). Syntax and semantics, in contrast, are less prone to variation and mostly rely on categorical information, which makes the communication between these modules and prosodic structure difficult. Nevertheless, the system introduced in this paper proves that an integration of spoken language into the existing computational grammars is possible and desirable in order to allow for a complete end-to-end analysis between form (the speech signal) and meaning (the semantic interpretation), and for an automatic analysis of linguistic phenomena from all relevant angles.

REFERENCES

R. Harald BAAYEN, Richard PIEPENBROCK, and Leon GULIKERS (1995), The CELEX lexical database (CD-ROM), in *Linguistic Data Consortium*, University of Pennsylvania, Philadelphia.

Anton BATLINER, Jan BUCKOW, Heinrich NIEMANN, Elmar NÖTH, and Volker WARNKE (2000), The prosody module, in Wolfgang WAHLSTER, editor, *Verbmobil: Foundations of speech-to-speech translation*, pp. 106–121, Springer, Berlin/Heidelberg.

Anton BATLINER, Bernd MÖBIUS, Gregor MÖHLER, Antje SCHWEITZER, and Elmar NÖTH (2001), Prosodic models, automatic speech understanding, and speech synthesis: Towards the common ground, in *Proceedings of the European Conference on Speech Communication and Technology*, volume 4, pp. 2285–2288, Aalborg, Denmark.

Michael BECKER, Joe PATER, and Christopher POTTS (2007), OT-Help: Java tools for Optimality Theory, http://web.linguist.umass.edu/~OTHelp/.

Kenneth R. BEESLEY and Lauri KARTTUNEN (2003), *Finite state morphology*, CSLI Publications, Stanford, CA.

Emily BENDER, Dan FLICKINGER, and Stephan OEPEN (2002), The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammar, in John CARROLL, Nelleke OOSTDIJK, and Richard SUTCLIFFE, editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pp. 8–14.

Steven BIRD (1992), Finite-state phonology in HPSG, in *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pp. 74–80.

Steven BIRD and Ewan KLEIN (1994), Phonological analysis in typed feature systems, *Computational Linguistics*, 20(3):455–491.

Daniel G. BOBROW, Bob CHESLOW, Cleo CONDORAVDI, Lauri KARTTUNEN, Tracy Holloway KING, Rowan NAIRN, Valeria DE PAIVA, Charlotte PRICE, and Annie ZAENEN (2007), PARC's Bridge and Question Answering System, in *Proceedings of the Grammar Engineering Across Frameworks Workshop (GEAF 2007)*.

Paul BOERSMA and David WEENINK (2021), Praat: doing phonetics by computer [computer program, Version 6.1.48], available at http://www.praat.org/.

Tina BÖGEL (2015), *The syntax-prosody interface in Lexical Functional Grammar*, Ph.D. thesis, University of Konstanz, Konstanz.

Tina BÖGEL (2020), German case ambiguities at the interface: Production and comprehension, in Gerrit KENTNER and Joost KREMERS, editors, *Prosody in syntactic encoding*, number 573 in Linguistische Arbeiten, pp. 51–84, De Gruyter, Berlin.

Tina BÖGEL (2023), Prosody and its interfaces, in Mary DALRYMPLE, editor, *Handbook of Lexical Functional Grammar*, Language Science Press, Berlin.

Tina BÖGEL, Miriam BUTT, Annette HAUTLI, and Sebastian SULGER (2007), Developing a finite-state morphological analyzer for Urdu and Hindi, in *Proceedings of the Sixth International Workshop on Finite-State Methods and Natural Language Processing (FSMNLP)*, Potsdam, Germany.

Tina BÖGEL and Alice TURK (2019), Frequency effects and prosodic boundary strength, in Sasha CALHOUN, Paola ESCUDERO, Marija TABAIN, and Paul WARREN, editors, *Proceedings of the 19th International Congress of Phonetic Sciences (ICPhS), Melbourne, Australia*, pp. 1014–1018.

Tina BÖGEL, Saeed Reza YOUSEFI, and Mahinnaz MIRDEHGHAN (2018), Vafsi oblique pronouns: stress-related placement patterns, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of LFG18*, pp. 88–108.

Kersti BÖRJARS, Rachel NORDLINGER, and Louisa SADLER (2019), *Lexical-Functional Grammar: An introduction*, Cambridge University Press, Cambridge.

Bettina BRAUN, Nicole DEHÉ, Jana NEITSCH, Daniela WOCHNER, and Katharina ZAHNER (2019), The prosody of rhetorical and information-seeking questions in German, *Language and Speech*, 62(4):779–807.

Norbert BRAUNSCHWEILER (2003), *Automatic detection of prosodic cues*, Ph.D. thesis, University of Konstanz.

Norbert BRAUNSCHWEILER (2006), The prosodizer – automatic prosodic annotations of speech synthesis databases, in *Proceedings of Speech Prosody*, Dresden, Germany, doi:10.21437/SpeechProsody.2006-136.

Joan BRESNAN, Ash ASUDEH, Ida TOIVONEN, and Stephen WECHSLER (2016), *Lexical-Functional Syntax*, Wiley-Blackwell, Malden, MA, 2 edition.

Miriam BUTT, Helge DYVIK, Tracy H. KING, Hiroshi MASUICHI, and Christian ROHRER (2002), The parallel grammar project, in *Proceedings of the COLING-2002 Workshop on Grammar Engineering and Evaluation*.

Miriam BUTT, Farhat JABEEN, and Tina BÖGEL (2020), Ambiguity resolution via the syntax-prosody interface: The case of *kya* 'what' in Urdu/Hindi, in Gerrit KENTNER and Joost KREMERS, editors, *Prosody in Syntactic Encoding*, volume 573 of *Linguistische Arbeiten*, pp. 85–118, De Gruyter, Berlin.

Miriam BUTT and Tracy Holloway KING (1998), Interfacing phonology with LFG, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of LFG98*, pp. 1–17, CSLI Publications, Stanford, CA.

Miriam BUTT, Tracy Holloway KING, María-Eugenia NIÑO, and Frédérique SEGOND (1999), *A grammar writer's cookbook*, CSLI Publications, Stanford, CA.

Stephen CLARK and James R. CURRAN (2007), Wide-coverage efficient statistical parsing with CCG and log-linear models, *Computational Linguistics*, 33(4):494–552.

Ann COPESTAKE (2002), *Implementing typed feature structure grammars*, CSLI Publications, Stanford, CA.

Richard CROUCH, Mary DALRYMPLE, Ronald M. KAPLAN, Tracy H. KING, John T. MAXWELL III, and Paula NEWMAN (2017), *XLE documentation*, Palo Alto Research Center, Palo Alto, CA, online documentation.

Mary DALRYMPLE, editor (2023), *The handbook of Lexical Functional Grammar*, number 13 in Empirically Oriented Theoretical Morphology and Syntax, Language Science Press, Berlin, doi:10.5281/zenodo.10037797.

Mary DALRYMPLE, John J. LOWE, and Louise MYCOCK (2019), *The Oxford reference guide to Lexical Functional Grammar*, Oxford University Press, Oxford.

Denys DUCHIER, Joseph LE ROUX, and Yannick PARMENTIER (2004), The metagrammar compiler: An NLP application with a multi-paradigm architecture, in *Proceedings of the 2nd Oz-Mozart conference, MOZ*, Charleroi.

Claire GARDENT and Yannick PARMENTIER (2005), Large scale semantic construction for tree adjoining grammars, in Philippe BLACHE, Edward STABLER, Joan BUSQUETS, and Richard MOOT, editors, *Proceedings of Logical Aspects of Computational Linguistics*, pp. 131–146, Bordeaux.

Anja GOLLRAD, Esther SOMMERFELD, and Frank KÜGLER (2010), Prosodic cue weighting in disambiguation: case ambiguity in German, in *Proceedings of Speech Prosody*, Chicago, doi:10.21437/SpeechProsody.2010-178.

Simon GONZALEZ, James GRAMA, and Catherine E. TRAVIS (2020), Comparing the performance of forced aligners used in sociophonetic research, *Linguistics Vanguard*, 6(1):1–13.

Martine GRICE and Stefan BAUMANN (2002), Deutsche Intonation und GToBI, *Linguistische Berichte*, 191:267–298.

Carlos GUSSENHOVEN (2008), Types of focus in English, in Chungmin LEE, Matthew GORDON, and Daniel BÜRING, editors, *Topic and focus*, volume 82 of *Studies in Linguistics and Philosophy*, pp. 83–100, Springer, Dordrecht.

Aravind JOSHI (2003), Tree-adjoining grammar, in Ruslan MITKOV, editor, *Handbook of computational linguistics*, Oxford University Press, Oxford.

Ronald M. KAPLAN and Joan BRESNAN (1982), Lexical-Functional Grammar: A formal system for grammatical representation, in Joan BRESNAN, editor, *The mental representation of grammatical relations*, pp. 173–281, MIT Press, Cambridge, MA.

Ronald M. KAPLAN, John T. MAXWELL III., Tracy Holloway KING, and Richard CROUCH (2004), Integrating finite-state technology with deep LFG grammars, in Erhard HINRICHS and Kiril SIMOV, editors, *Proceedings of the ESSLLI Workshop on Combining Shallow and Deep Processing for NLP*, pp. 11–20.

Thomas KISLER, Uwe REICHEL, and Florian SCHIEL (2017), Multilingual processing of speech via web services, *Computer Speech and Language*, 45:326–347.

Ewan KLEIN (2000), A constraint-based approach to English prosodic constituents, in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 217–224.

Manfred KRIFKA (2008), Basic notions of information structure, *Acta Linguistica Hungarica*, 55(3–4):243–276.

D. Robert LADD (1984), Declination: A review and some hypotheses, *Phonology*, 1:53–74.

Géraldine LEGENDRE, Yoshiro MIYATA, and Paul SMOLENSKY (1990), Harmonic grammar – a formal multi-level connectionist theory of linguistic well-formedness: An application, in *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, pp. 388–395, Lawrence Erlbaum, Cambridge, MA.

Ilse LEHISTE, Joseph P. OLIVE, and Lynn A. STREETER (1976), Role of duration in disambiguating syntactically ambiguous sentences, *The Journal of the Acoustical Society of America*, 60:1199–1202.

Willem J.M. LEVELT, Ardi ROELOFS, and Antje S. MEYER (1999), A theory of lexical access in speech production, *Behavioral and Brain Sciences*, 22:1–75.

Moritz MESSMER and Mark-Matthias ZYMLA (2018), The Glue Semantics work bench: A modular toolkit for exploring Linear Logic and Glue Semantics, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of LFG18*, pp. 268–282, CSLI Publications, Stanford, CA.

Gerald PENN and Bob CARPENTER (1999), ALE for speech: A translation prototype, in *Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH'99)*, pp. 947–950, Budapest, Hungary.

Karin PITTNER (2014), Ist der Dativ dem Genitiv sein Tod? Funktionen und Konkurrenzformen von Genitiv-NPs im heutigen Deutsch (Is the dative the death of the genitive? The function and competition of Genitive-NPs in Modern German), in Corinna REUTER and Anne-Kathrin SCHLIEF, editors, *Linguistische und sprachdidaktische Aspekte germanistischer Forschung Chinesisch-Deutsch*, pp. 41–56, Peter Lang, Frankfurt am Main.

Carl J. POLLARD and Ivan A. SAG (1994), *Head-driven Phrase Structure Grammar*, University of Chicago Press, Chicago.

Chris POTTS, Joe PATER, Karen JESNEY, Rajesh BHATT, and Michael BECKER (2010), Harmonic grammar with linear programming: From linear systems to linguistic typology, *Phonology*, 27(1):77–117.

Patti PRICE, Mari OSTENDORF, Stefanie SHATTUCK-HUFNAGEL, and Cynthia FONG (1991), The use of prosody in syntactic disambiguation, *Journal of the Acoustical Society of America*, 90(6):2956–2970.

Alan PRINCE and Paul SMOLENSKY (2004), *Optimality Theory: Constraint interaction in generative grammar*, Blackwell, Oxford, Malden, MA.

R CORE TEAM (2016), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

Mats ROOTH (2016), Alternative semantics, in Caroline FÉRY and Shinichiro ISHIHARA, editors, *Oxford handbook of information structure*, pp. 19–40, Oxford University Press, Oxford.

Florian SCHIEL (1999), Automatic phonetic transcription of non-prompted speech, in John J. OHALA, Yoko HASEGAWA, Manjari OHALA, Daniel GRANVILLE, and Ashlee C. BAILEY, editors, *Proceedings of the 14th International Congress of Phonetic Sciences (ICPhS), San Fracisco, CA, USA*, pp. 607–610, San Francisco.

Antje SCHWEITZER and Bernd MÖBIUS (2009), Experiments on automatic prosodic labeling, in *Proceedings of INTERSPEECH*, pp. 2515–2518, Brighton, UK.

Alan K. SCOTT (2011), Everyday language in the spotlight: The decline of the genitive case, *German as a Foreign Language*, 1(1):53–70.

Elisabeth O. SELKIRK (2011), The syntax-phonology interface, in John GOLDSMITH, Jason RIGGLE, and Alan YU, editors, *The handbook of phonological theory*, pp. 435–484, Blackwell, Malden, MA.

Kim SILVERMAN, Mary BECKMAN, John PITRELLI, Mari OSTENDORF, Colin WIGHTMAN, Patti PRICE, Janet PIERREHUMBERT, and Julia HIRSCHBERG (1992), ToBI: A standard for labeling English prosody, in *Proceedings of the 1992 International Conference on Spoken Language Processing*, pp. 867–870, Banff.

Mark STEEDMAN (2000), *The syntactic process*, MIT Press, Cambridge, MA.

Sebastian SULGER, Miriam BUTT, Tracy Holloway KING, Paul MEURER, Tibor LACZKÓ, György RÁKOSI, Cheikh Bamba DIONE, Helge DYVIK, Victoria ROSÉN, Koenraad DE SMEDT, Agnieszka PATEJUK, Ozlem CETINOGLU, I Wayan ARKA, and Meladel MISTICA (2013), Pargrambank: The pargram parallel treebank, in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 550–560, Association for Computational Linguistics, Sofia, Bulgaria.

Bruce TESAR and Paul SMOLENSKY (1998), Learning optimality-theoretic grammars, *Lingua*, 106:161–196.

Hubert TRUCKENBRODT (1999), On the relation between syntactic phrases and phonological phrases, *Linguistic Inquiry*, 30(2):219–255.

Wolfgang WAHLSTER (2013), *Verbmobil: Foundations of speech-to-speech translation*, Springer, Berlin/Heidelberg.

John C. WELLS (1997), SAMPA computer readable phonetic alphabet, in Dafydd GIBBON, Roger MOORE, and Richard WINSKI, editors, *Handbook of standards and resources for spoken language systems*, pp. 684–732, Mounton de Gruyter, Berlin, New York.

Bodo WINTER (2019), *Statistics for linguists: An introduction using R*, Routledge, New York London.

Ian H. WITTEN and Eibe FRANK (2005), *Data mining: Practical machine learning tools and techniques*, Morgan Kaufmann, San Francisco, USA, 2nd edition.

Yi Xu (2005), Speech melody as articulatorily implemented communicative functions, *Speech Communication*, 46:220–251.

Yi Xu (2013), ProsodyPro – A tool for large-scale systematic prosody analysis, in Brigitte Bigi and Daniel Hirst, editors, *Tools and resources for the analysis of speech prosody*, pp. 7–10, Laboratoire Parole et Langage, France.

Yi Xu and Ching X. Xu (2005), Phonetic realization of focus in English declarative intonation, *Journal of Phonetics*, 33:159–197.

Kristine M. Yu (2018), Advantages of constituency: Computational perspectives on word prosody in Samoan, in Annie Foret, Reinhard Muskens, and Sylvain Pogodalla, editors, *Formal Grammar (FG 2017)*, pp. 105–124, Springer, Berlin/Heidelberg.

Marzena Żygis, John M. Tomlinson, Caterina Petrone, and Dominik Pfütze (2019), Acoustic cues of prosodic boundaries in German at different speech rates, in Sasha Calhoun, Paola Escudero, Marija Tabain, and Paul Warren, editors, *Proceedings of the 19th International Congress of Phonetic Sciences (ICPhS), Melbourne, Australia*, https://hal.science/hal-02097707.

Tina Bögel
0000-0001-5644-3730
Tina.Boegel@uni-konstanz.de

Department of Linguistics
Universitätsstraße 10,
78464 Konstanz,
Germany

Tianyi Zhao
0009-0000-5258-4069
Tianyi.Zhao@uni-konstanz.de

Department of Linguistics
Universitätsstraße 10,
78464 Konstanz,
Germany

Tina Bögel and Tianyi Zhao (2025), *From speech signal to syntactic structure: A computational implementation*, Journal of Language Modelling, 13(1):1–42
https://dx.doi.org/10.15398/jlm.v13i1.397

Tier-based strict locality and the typology of agreement

Kenneth Hanson
Stony Brook University

ABSTRACT

This paper presents a subregular analysis of syntactic agreement patterns modeled using *command strings* over Minimalist Grammar (MG) dependency trees (Graf and Shafiei 2019), incorporating a novel MG treatment of agreement. Phenomena of interest include relativized minimality and its exceptions, direction of feature transmission, and configurations involving chains of agreeing elements. Such patterns are shown to fall within the class of *tier-based strictly 2-local* (TSL-2) languages, which has previously been argued to subsume the majority of long-distance syntactic phenomena, as well as those in phonology and morphology (Graf 2022a). This characterization places a tight upper bound on the range of configurations that are predicted to occur while providing parameters for variation which closely match the observed typology.

Keywords: syntax, agreement, locality, Minimalist Grammars, tier-based strictly local languages

1 INTRODUCTION

Linguistic patterns display tremendous variation, yet are also subject to strong structural constraints. For example, syntactic dependencies are generally understood to follow the *c-command* relation (Reinhart 1981) and also to obey *relativized minimality* (Rizzi 1990). But this is not the full picture. In recent years, it has become increasingly apparent that whatever mechanism underlies agreement must be parameterized so as to make fine-grained distinctions regarding which elements agree, for what features, and under what configurations (Bobaljik 2008; Deal 2015; Keine 2019, a.o.). From a computational perspective, what is striking about the range of attested patterns is that, under the appropriate representation, they largely fall within the class of *tier-based strictly local* (TSL) languages, mirroring previous results on long-distance phonotactics (McMullin 2016), movement (Graf 2022b) and case (Vu *et al.* 2019; Hanson 2023). The primary aim of this paper is to demonstrate this.

The basic intuition behind a TSL pattern is that when the irrelevant elements are ignored, the pattern can be described using local constraints on those that remain visible. To illustrate, consider the standard Minimalist treatment of subject-verb agreement. Finite T bears unvalued ϕ-features which serve as a probe for agreement, and the subject DP bears valued ϕ-features which serve as a goal. The probe must c-command the goal; additionally, the probe and goal may occur at some distance from each other as long as no other ϕ-bearing element intervenes (i.e. it must obey relativized minimality). If we take the chain of elements along the clausal spine below T and ignore everything except for these elements, we obtain a string called a *tier projection*. The relevant local constraint on the tier is that T must agree with the immediately following D. This is schematized below:

(1) $T_{[\phi]} \dots D_{[\phi]} \dots D_{[\phi]}$ $\qquad$ $T_{[\phi]} \dots D_{[\phi]} \dots D_{[\phi]}$

Crucially, TSL is a restrictive class of formal languages. Aside from the fact that it can relate elements at a distance, the space of patterns that it can express is severely limited. This helps to explain why we see the patterns that we do and no others. In fact, the range of linguistic patterns which have been described as being TSL are overwhelmingly TSL-2, meaning that all constraints can be stated within a window of two elements on the tier. Here, I show that the parameters provided by the formalism – the set of elements which appear on the tier and the local constraints on the tier – closely correspond to several key dimensions of the formal typology of agreement, echoing previous results on movement (Graf 2022b) and long-distance harmony

(McMullin and Hansson 2016). The parallel between syntactic agreement and phonological harmony is particularly striking; this observation is not entirely new (cf. Nevins 2010), but the present perspective brings it into unusually sharp focus. Overall, these results lend further support to the idea that linguistic phenomena across domains are united in the kinds of computations they are built upon (Graf 2022a).

In order to show that agreement patterns are TSL-2, I adopt a formalization based on *Minimalist Grammars* (MGs, Stabler 1997, 2011) and *command-strings* (c-strings, Graf and Shafiei 2019), which are paths through an MG dependency tree whose ordering corresponds approximately to asymmetric c-command. In doing this, a novel approach to agreement is proposed, which utilizes “probe” and “goal” features analogous to standard MG licensor and licensee features. In addition to highlighting the parallel between syntax and phonology, this model also allows us to cleanly separate phenomena which are explained well by computational restrictions from those which derive from other sources such as the tree geometry; the latter include the c-command restriction as well as certain island constraints.

Similarly, some typological generalizations most likely derive from extragrammatical factors, such as constraints on language acquisition. While the formalism does not say anything about which sets of elements may form a tier, it is natural to suppose that the learner only considers elements which are obviously related. There is evidence from phonotactic learning that this is indeed the case, as participants in an artificial language experiment succeeded at learning long-distance dependencies involving only consonants or vowels, but failed to learn those involving both (Newport and Aslin 2004). Exceptions to this rule could occur when agreeing DPs differ from non-agreeing DPs in a highly salient manner, such as bearing a specific case or undergoing movement, as discussed in detail here.

The remainder of this paper is structured as follows. In Section 2, I provide an overview of the classification of linguistic patterns according to their computational complexity, and introduce the formal class TSL along with its subclass SL (strictly local). Section 3 develops a formal model which allows agreement patterns to be analyzed using TSL-2 constraints over c-strings, and presents an analysis of subject-verb agreement. In Section 4, I show that a wide variety of agreement patterns across languages are just slight variations of the basic TSL-2 pattern. Section 5 concludes, with a discussion of some open questions.

2 COMPUTATIONAL COMPLEXITY

In this section, we review some concepts from formal language theory which are crucial to the TSL analysis of linguistic patterns. I start with the motivation for modeling syntactic dependencies with subregular constraints over trees (Section 2.1). From there, I provide definitions and examples of SL (Section 2.2) and TSL (and Section 2.3), the two classes most relevant to local and long-distance linguistic syntactic patterns, followed by a brief discussion of multi-TSL grammars, which represent the intersection of multiple (T)SL constraints (Section 2.4).

2.1 *String languages and tree languages*

Formal languages are sets of objects, traditionally strings, which can be used to model linguistic patterns. We can categorize the complexity of formal languages in terms of the kinds of patterns they are able to represent – more complex classes of languages can encode a wider range of patterns. Generally speaking, more complex classes also require more powerful machinery both to learn and process. *String languages* (or stringsets) are commonly used to model the computational complexity of phonological and morphological patterns, but for syntax *tree languages* (or treesets) are more insightful. While the reasons for this are intuitive to syntacticians, we can also motivate this representational choice from a purely formal perspective.

The Chomsky Hierarchy (Chomsky 1959), shown in Figure 1, outlines the major classes of string languages.[1] Phonological and mor-

[1] Many details are omitted from this figure for simplicity. For example, syntax does not seem to require the full power of the context-sensitive languages, but rather some *mildly context-sensitive* subclass (Joshi 1985). Furthermore, only the Swiss German data in Shieber 1985 is not context-free on the surface, unlike the earlier Dutch data in Huybregts 1976, 1984. However, Bresnan *et al.* (1982) argue that when considering the structures assigned to sentences by the grammar, Dutch is also not context-free. Thus, the importance of structures emphasized in this section has a long history.

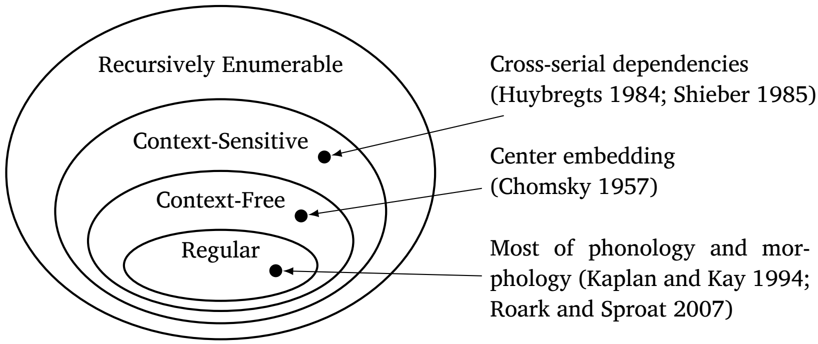


Figure 1: The original Chomsky Hierarchy (Chomsky 1959), showing the categorization of important linguistic patterns when modeled as surface strings

phological patterns almost exclusively lie within the class of *regular languages*, while many syntactic patterns (analyzed as surface strings) are *context-free*, and some are *context-sensitive*. While useful in many respects, this characterization also obscures the formal similarities between phonology and morphology on the one hand, and syntax on the other. The classification of syntax is particularly problematic, for there are many types of *regular* patterns which are not attested in any module of grammar, including syntax.

In recent years a more fine-grained view has emerged. We can decompose the regular languages of the Chomsky Hierarchy into many smaller classes of *subregular languages*. A relevant subset of the resulting Subregular Hierarchy is shown in Figure 2. Many of these classes have been known for some time (Schützenberger 1965; McNaughton and Papert 1971; Simon 1975), but their significance for language cognition has only been recognized more recently (Rogers *et al.* 2013; Heinz 2018; Graf 2022a). It is now conjectured that all

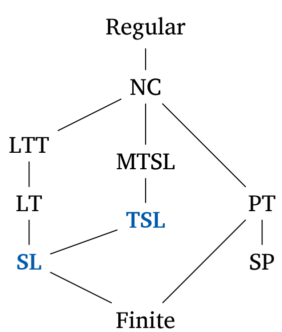


Figure 2: The (simplified) Subregular Hierarchy, adapted from Heinz 2018. SL and TSL, which subsume most phonological and morphological phenomena, are highlighted. By hypothesis, syntactic dependencies fall under the tree-based equivalents of SL and TSL

linguistic patterns lie within the subregular region: phonological and morphological patterns are subregular string languages, while syntactic patterns fall in the tree-based equivalents of these classes. This has been termed the *cognitive parallelism hypothesis* by Graf (2022a).

Following recent work, I pursue the specific hypothesis that local dependencies fall within the *strictly local* (SL) languages, while long-distance dependencies are *tier-based strictly local* (TSL), which is a proper superclass of SL. Within syntax, the former includes selection (Graf 2018) and functional hierarchies (Hanson 2024a), while the latter includes movement (Graf 2018, 2022b) and case (Vu *et al.* 2019; Hanson 2023). Adding to this, I argue that agreement is also TSL.

If this is correct, then we have an explanation for why linguistically preposterous constraints along the lines of "a sentence may not contain both a verb and an adjective unless it also contains at least one quantifier" and "a word must not include both a consonant cluster and vowel hiatus" do not exist: such constraints are LT (locally testable), but not TSL, and therefore not of the variety handled by the computational machinery underlying language. Similarly non-existent patterns include "a sentence must contain between two and four adverbs", which is LTT (locally threshold testable), "a word must obey consonant harmony or vowel harmony, but need not obey both", which is NC (non-counting, a.k.a. star-free), and "the number of prepositions in a sentence must be a multiple of three", which is properly regular. In addition, TSL languages can be learned in polynomial time and data (Jardine and McMullin 2017; Lambert 2021), and in stark contrast to the classes just mentioned, with low memory requirements (Lambert *et al.* 2021). As a consequence, the typological facts that we attribute to computational complexity might ultimately be grounded in considerations of efficient learnability (ibid.).

Note that while some linguistic patterns do in fact go beyond TSL, only a few have been noticed in the literature so far, and crucially, none occur in the data examined here.[2] Furthermore, as discussed in detail in Section 4, the typology of agreement patterns fits very closely

[2] Known examples include Samala sibilant harmony, Uyghur backness harmony, and Sanskrit n-retroflexion. All of these can be modeled with a set of extensions to TSL known collectively as *structure-sensitive* TSL (cf. De Santo and Graf 2019; Mayer and Major 2018; Graf and Mayer 2018).

to what we predict based on the expressive capabilities of TSL with a size two window. Because of this, I describe only SL and TSL in detail.

Strictly local languages 2.2

Each class of subregular languages has several equivalent characterizations. Here, I present definitions using *forbidden factors* – which for our purposes may be either substrings (as in SL) or substrings on a tier (as in TSL) – adapted from Mayer 2021. For illustration, I draw on examples from phonotactics.

In what follows, S^n and S^* denote all strings over set S of length n and of any finite length, respectively. Also, s^k denotes the string consisting of k repetitions of s. Σ denotes a finite set of symbols called the *alphabet*, and a *string language* is a subset of Σ^*.

Intuitively, a SL grammar is just a finite set of forbidden substrings of some fixed length, and the corresponding SL language is the set of all strings that do not contain any forbidden substrings.[3] We formalize this intuition as follows. $\rtimes$ and $\ltimes$ are the *edge markers*, which are added to a string so that the beginning, middle, and end can be modeled uniformly. Next, the *k-factors* of a string w, denoted $f_k(w)$, are the set of length-k substrings of $\rtimes^{k-1}w\ltimes^{k-1}$. For example, the 2-factors of the string *abcabc* are $\{\rtimes a, ab, bc, ca, c\ltimes\}$; the string *abcabcabc* also contains the same 2-factors. A grammar containing the 2-factor *ab* would rule out both of these strings, along with many others.

DEFINITION 1 *A* strictly k-local *(SL-k) grammar is a finite set* $G \subseteq (\Sigma \cup \{\rtimes, \ltimes\})^k$. *A language* $L \subseteq \Sigma^*$ *is SL-k iff there exists a SL-k grammar G such that*

$$L = \{w \in \Sigma^* : f_k(w) \cap G = \emptyset\}$$

A language is SL iff it is SL-k for some $k \geq 1$.

SL grammars correspond to categorical k-gram models, and express what a linguist would identify as local constraints. As a simple example, consider a natural language which exhibits CV syllable

[3] Equivalently, we can define a SL grammar as a set of *permissible* substrings. Converting between these two formulations is trivial, assuming that all substrings in the grammar have the same length and that wellformedness is categorical.

structure with an optional word-final consonant. We can model this pattern as a SL language consisting of strings of the symbols C and V, as summarized in (2). Licit words in this language include CV, CVC, and CVCV, but not VC, CVV, or CVCCV. The forbidden substrings for this language are {⋊V, VV, CC}, making it SL-2. The licit word ⋊CVC⋉ contains the substrings {⋊C, VC, VC, C⋉}, none of which are forbidden. On the other hand, the illicit word ⋊VC⋉ contains {⋊V, VC, C⋉}, of which ⋊V is forbidden.

(2) Example SL-2 pattern: CV syllable structure, optional final C

- $\Sigma = \{C, V\}$
- $k = 2$
- $G = \{⋊V, VV, CC\}$
- Licit words: ⋊CV⋉, ⋊CVC⋉, ⋊CVCV⋉, ⋊CVCVC⋉, …
- Illicit words: ⋊V⋉, ⋊VC⋉, ⋊CVV⋉, ⋊CVCCV⋉, …

Note that SL grammars cannot relate two symbols that do not occur within the same k-factor, nor can they count occurrences of k-factors; if two strings contain the same set of k-factors, they are indistinguishable. These restrictions distinguish SL (and TSL) from more powerful classes in the subregular hierarchy. Note also that a grammar which enforces this abstract constraint using separate symbols for each distinct consonant and vowel, while considerably *larger*, is no more *complex* in the relevant sense.

2.3 *Tier-based strictly local languages*

A TSL language is much like a SL language except that the forbidden factors are substrings on a tier, allowing a limited type of long-distance dependency to be expressed (Heinz *et al.* 2011; Lambert and Rogers 2020). Any symbol not appearing on the tier is ignored completely and the remainder are treated as adjacent; a SL language is the special case of TSL where every symbol appears on the tier. Note that while this notion of a tier was inspired by autosegmental phonology (Goldsmith 1976), it is conceptually distinct, as the tier elements are just a special subset of the elements of the full structure. Lambert (2023) uses the term *relativized adjacency* to describe the type of relativized locality encapsulated by a tier in this sense.

Formally, in addition to the alphabet Σ, there is also a *tier alphabet* T, and every string w is associated with a *tier projection*, denoted $\text{PROJ}_T(w)$, in which all symbols not in T are removed. A string is in the language iff its tier projection contains no forbidden k-factors.

DEFINITION 2 *A* tier-based strictly k-local *(TSL-k) grammar is a tuple* (T, G) *where* $T \subseteq \Sigma$ *and* $G \subseteq (T \cup \{\rtimes, \ltimes\})^k$. *A language* L *is TSL-*k *iff there exists a TSL-*k *grammar such that*

$$L = \{w \in \Sigma^* : f_k(\text{PROJ}_T(w)) \cap G = \emptyset\}$$

A language is TSL if it is TSL-k *for some* $k \geq 1$.

For our next example, consider a language with (symmetric) sibilant harmony, a TSL-2 pattern, as shown in (3). Licit words in such a language include 'saksa' and 'ʃakʃa', but not 'sakʃa' or 'ʃaksa'. In this case, the tier alphabet contains only the sibilant consonants {s, ʃ}. Mismatched sibilants are forbidden on the tier, ruling out any strings that do not obey harmony. For example, the illicit word 'sakʃa' has the tier projection ⋊sʃ⋉, which contains the forbidden substring sʃ.

(3) Example TSL-2 pattern: Sibilant harmony

- Σ = {a,k,s,ʃ}
- k = 2
- T = {s,ʃ}
- G = {sʃ,ʃs}
- Licit words: asa, aʃa, saksa, ʃakʃa, …
- Illicit words: saʃa, ʃasa, sakʃa, ʃaksa, …

While the pattern just described is also in the class SP (strictly piecewise, Rogers *et al.* 2010), and some harmony patterns are SL with a suitably large k-value, only TSL-2 subsumes both types as well as long-distance harmony with blocking (cf. McMullin and Hansson 2016). The latter type is particularly pervasive in syntax, making TSL-2 the prime candidate for the maximally restrictive classification of long-distance syntactic patterns.

Multi-tier grammars 2.4

The reader may have noticed that in (3) there is nothing preventing the generation of absurd words such as *skskaaakkk*. To obtain a full model, we must intersect the subregular languages representing isolated patterns like those above to produce one that obeys all of them.

The intersection of several TSL languages is known as a multi-TSL, or MTSL language (De Santo and Graf 2019). In general, a realistic description of any natural language is necessarily (at least) MTSL due to the existence of both local and long-distance dependencies. Furthermore, it is empirically well-established that long-distance dependencies such as EPP movement, *wh*-movement, ϕ-agreement are subject to different locality constraints (cf. Keine 2019). Thus, when we say that long-distance dependencies are in general TSL, this should be interpreted to mean that each individual dependency is TSL.

This raises the question of what exactly constitutes an independent linguistic dependency. For example, the analysis of case in Japanese by Hanson (2023) includes three tiers, and is therefore technically MTSL. The same situation is likely to arise in agreement patterns in which a single predicate agrees with multiple noun phrases simultaneously, such as those analyzed by Béjar and Rezac (2009) and Nevins (2011). It seems plausible that such patterns can likewise be decomposed into a set of intersecting constraints, each of which is TSL. However, it could be the case that interactions other than intersection are needed, in which case the full pattern is not MTSL. Due to the complexity of the data, a proper investigation of this issue is beyond the scope of the present article, which focuses exclusively on "individual" dependencies.

3 A TSL MODEL OF AGREEMENT

In this section, we extend TSL languages to trees in order to model agreement. Following recent work (Graf and Shafiei 2019; Graf 2022b; Hanson 2023), I use Minimalist Grammar dependency trees (Sections 3.1 and 3.2) for the tree language. To date, there are two ways in which TSL languages have been generalized to trees. Here, I develop a model based on command strings (Section 3.3), and show how the model can be applied to agreement (Section 3.4).

Minimalist Grammars 3.1

Minimalist Grammars (MGs, Stabler 1997, 2011) are a formalization of ideas from Chomsky's (1995) Minimalist Program. Standard MGs contain just two operations: Merge and Move (we will add agreement later). The grammar of a language is just a lexicon of syntactic heads annotated with features to guide these operations.

The features for each operation come in two polarities, which I notate +F and −F. For Merge, these are *selector* and *category* features, whose meanings are intuitive. For example, transitive *v* has selector features +V and +D, since it selects a VP complement and a DP specifier, and a category feature −*v*. Since these features play no direct role in this paper and can be inferred from context, I omit them in all derivations. Additionally, I will continue to refer to categories as V/D/C, etc., even though they are technically −V/−D/−C. For Move, we have *licensor* features, which mark the landing site of movement, and *licensee* features, which mark the mover. For example, finite T carries +EPP, and the DP which moves to its specifier carries −EPP.[4]

It is important to note that MG features are just diacritics which describe what happens in the derivation. In a language with single *wh*-movement, for example, only the highest *wh*-element bears −Wh since it is the one that must move. Placing −Wh on every *wh*-element is tantamount to saying that all of them move; indeed, this is what Graf and Kostyszyn (2021) do in their model of multiple *wh*-movement.

MGs can be used to generate syntactic structures in several ways. The standard approach is to generate a language of *derivation trees*, which show the order of Merge and Move steps. The derivation tree is then mapped to a phrase structure tree by executing all movements and inserting X′-style labels. It is the derivation tree language which is our focus, as this is where syntactic dependencies are formed. The constraints on the mapping to the derived tree are also a topic of current

[4] In Stabler's original notation selector/category/licensor/licensee features are notated =f/f/+f/−f. In Graf 2018 and related work they are $F^+/F^-/f^+/f^-$. I selected the present notation in part because the addition of agreement features in Section 3.4 produces a six-way distinction. For our purposes it is unlikely that Merge/Move/Agree features will be misinterpreted, so a binary split is sufficient.

research; see Graf 2023 for a subregular model which handles placement of moved elements in the correct position.

3.2 *Dependency trees*

The specific kind of derivation tree we will use is a *dependency tree*. This representation is especially compact while providing all necessary information about the derivation, namely what elements merged with what, and what their features are.[5]

An example phrase structure tree and the corresponding dependency tree for the sentence “The cat chases the rats” are shown in Figure 3. Every node in the dependency tree is a feature-annotated

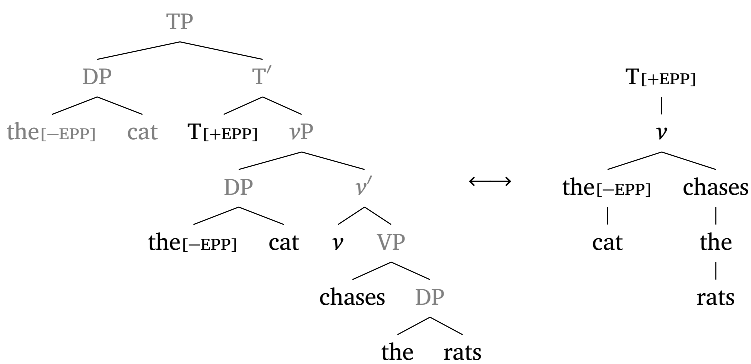


Figure 3: Phrase structure tree and dependency tree for “The cat chases the rats”. Nodes appearing only in the phrase structure tree are grayed out

lexical item, but as mentioned above I omit all selector and category features for brevity. The daughters of a node are the heads of its arguments, ordered from right to left in order of first merge. Thus, the rightmost daughter is the complement and any others are specifiers. For example, the left daughter of *v* is the determiner heading its specifier, while the right daughter is the verb *chases*, which heads the complement. Additionally, all nodes in the dependency tree appear in base

[5] These MG dependency trees first appear in Graf and Shafiei 2019 and are formally defined in Graf and Kostyszyn 2021. However, the use of dependency structures in MG has extensive precedent. The earliest use seems to be Kobele 2002, and the system in Kobele 2012 is essentially identical to that used here. Also see Boston *et al.* 2010, who use MGs to derive surface dependency trees.

position only. In the present example, this applies to the subject, which undergoes EPP movement to Spec-TP.

Our goal is to show that the set of licit feature configurations conforms to a TSL grammar over dependency trees. There are several ways in which this can be done. Graf (2018) defines a direct analog of string TSL: the tiers are trees, and the constraints restrict the string of daughters of each node on the tier. Graf and Shafiei (2019) propose an alternative in which we extract paths through the derivation tree along which syntactic dependencies occur and enforce constraints on the resulting string language. I adopt a modified version of the latter approach, as described below.

As a final note, although elements appear only in their base position in the dependency tree, it is often nonetheless possible to handle interactions with movement just by inspecting the features of the moving elements. For example, differential object marking in many languages can be analyzed as being fed by movement out of VP, as in the analysis of Sakha by Baker and Vinokurova (2010). In this case, presence or absence of a particular licensee feature on the D head of the object is enough to determine if it should be marked. This method will not work when it is crucial to know the exact landing site, but it will work whenever we just need to know whether or not a phrase has moved at all and perhaps also the type of movement, as is true of the patterns examined in this paper.

3.3 *Command strings and spines*

The specific model I utilize in this paper, building upon ideas in Graf and De Santo 2019, splits each tree into a set of strings, which represent the *complement spine* of the tree and each complex left branch. But first, let us overview the basics of *command strings* (c-strings) as introduced by Graf and Shafiei (2019).

The c-string of a node is the path to that node from the root which includes its left siblings, its ancestors, and the left siblings of its ancestors, with both sibling and ancestor order preserved. Put another way, every path from the root made by tracing the first-daughter and left-sister relations is a c-string. This is schematized on the left side of

Figure 4:
Left: a c-string follows the mother-of relation to the first daughter (filled arrow) and the left-sister relation (open arrow). The mother-of relation to non-first daughters (dotted lines) is not used. Right: c-string for *rats* in the sentence 'The cat chases the rats' (blue dashed lines)

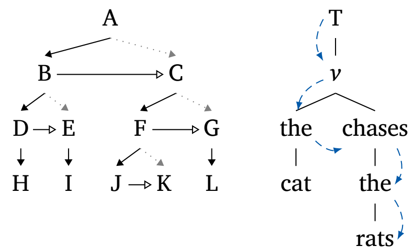


Figure 4. As a concrete example, the c-string for *rats* in our running example sentence is $\mathrm{T}_{[+\text{EPP}]} \cdot v \cdot \mathrm{the}_{[-\text{EPP}]} \cdot \mathrm{chases} \cdot \mathrm{the} \cdot \mathrm{rats}$, as illustrated on the right side of the same figure.

A formal definition is given below. This definition is identical to that given by Graf and Shafiei except that the ordering is from root to target node rather than the reverse. This allows c-strings to be read more easily, but is otherwise inconsequential since TSL string languages are closed under reversal, as are many other subregular classes.

DEFINITION 3 *Let T be a tree such that node m has the daughters $d_1, \ldots, d_i, \ldots, d_n$ with $n \geq 0$. The immediate c-string $ics(d_i)$ of d_i is the string $d_1 \cdots d_i$. For every node n of T, its c-string $cs(n)$ is recursively defined as follows, where $\cdot$ indicates string concatenation:*

$$cs(n) := \begin{cases} n & \text{if } n \text{ is the root of } T \\ cs(m) \cdot ics(n) & \text{if } m \text{ is } n\text{'s mother} \end{cases}$$

The ordering relation encoded by a c-string, which Graf and Shafiei call *d(erivational)-command*, reflects the hierarchical order of maximal projections in the phrase structure tree, or alternatively, the order in which category features are checked. It can be thought of as a hybrid of asymmetric c-command, since the complement is commanded by both its head and the specifier, and m-command (Aoun and Sportiche 1982), since the head commands the specifier rather than the other way around. For example, T d-commands its complement *v*, which in turn d-commands V; in addition, *v* d-commands the D which heads its specifier. Also notice that, since d-command is defined in terms of the dependency tree, it avoids unnecessary complications related to X′ projections and their labels. For example, it rarely matters whether the head of XP commands its specifier or the other way around, but the maximal projection should certainly be considered

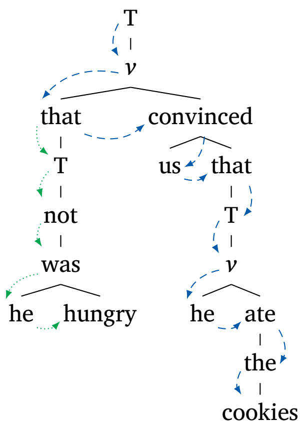


Figure 5:
Tree and two spines for the sentence "[That he was not hungry] convinced us [that he ate the cookies]". The subject clause begins a new spine (green dotted), while the complement clause shares a spine with the main clause (blue dashed)

superior to the specifier. D-command provides the latter, as desired. In the end, this allows us to enforce both traditional c-command-based constraints, such as reflexive licensing, as well as containment-based constraints, such as islands, in a simple and unified manner.

Now, Graf and Shafiei are concerned primarily with licensing of individual nodes; for them, wellformedness of a dependency tree requires (among other things) that all c-strings are well-formed.[6] In this paper, I utilize just the c-string which traces the *complement spine* (henceforth *spine*) of the tree, plus the partial c-strings which trace the spine of some complex left branch. Figure 5 gives an example with two spines: *[That he was not hungry] convinced us [that he ate the cookies]*. The complement clause is part of the main spine (T · *v* · that · convinced · us · that · T · *v* · he · ate · the · cookies), while the subject clause constitutes its own spine (that · T · not · was · he · hungry). Notice that each node appears in at most two spines: those which head a complex left branch appear in both the spine of that branch and in the containing spine. In the present example, this applies to *that* in *that he was not hungry*.

From now on, I will refer to the (partial) c-strings tracing spines as *spinal* c-strings, and all of our TSL grammars will apply to these strings. This will allow us to model pairwise dependencies such as agreement

[6] As will be discussed momentarily, not all syntactic constraints can be modeled with c-strings. For example, as a reviewer remarks, they cannot enforce the SMC of standard MGs; see Graf and Kostyszyn 2021 for a TSL treatment.

in manner which is highly intuitive and which closely parallels the treatment of phonological harmony discussed previously. However, there is an extra benefit to making this shift. As discussed by Graf and De Santo (2019), syntactic dependencies tend not to occur between a head and an element deeply embedded in some adjunct or specifier. For example, movement out of adjuncts and specifiers is often degraded (these are the well-known *adjunct island* and *specifier island* constraints), while movement out of complements is unremarkable. Similarly, finite T usually agrees with a DP in the same spine, perhaps embedded in a complement clause, but not one which is embedded in a subject clause. As a consequence, by applying our grammars to spinal c-strings, we effectively enforce these constraints as well.

There are, of course, numerous exceptions to this generalization, such as reflexive licensing (in which information must be passed down all paths) and parasitic gaps (which cannot be handled with c-strings at all). Additionally, since c-strings conflate ancestors and c-commanders, there are situations where constraints which should only affect one type of blocker are incorrectly applied to both; only containing phrases should induce island effects, for instance. Enriching the c-string representation to indicate whether the path has just entered a specifier/complement/adjunct, as in Graf and De Santo 2019, would allow some of these complications to be handled directly. However, tackling such issues here would take us too far afield, so I leave the development of a more complete model for future research.

3.4 *Constraining agreement*

Having established how to obtain a spinal c-string from a dependency tree, we are almost ready to show how TSL constraints on these strings can be used to model syntactic agreement. But first, recall that standard MGs have no agreement operation; we now add one.

As mentioned in the introduction, I assume agreement to involve two types of features: *unvalued* features, which receive their value during the derivation and *valued* features, which enter the derivation with their value; I will refer to a node with unvalued features as a *probe*, and the node which provides its value as a *goal*. Accordingly, I define a notation for agreement probes and goals parallel to the movement

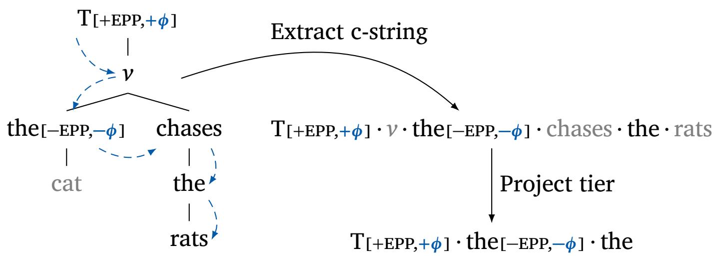


Figure 6: TSL analysis of English subject-verb agreement. Elements ignored at each step are grayed out

features of standard MGs. For each agreement feature F, +F denotes a probe and −F denotes a goal. In the case of subject-verb agreement, T bears $+\phi$ and the D head that it agrees with bears $-\phi$.[7] To be clear, this does *not* mean that other heads do not bear ϕ-features in the theoretical sense, only that they do not serve as the goal of agreement in the current derivation. This is analogous to the MG treatment of movement: a *potential* EPP mover (or *wh*-mover, etc.) does not necessarily bear −EPP (or −Wh), only *actual* movers do.[8]

Now, we restrict the set of English dependency trees to just those with well-formed agreement feature configurations. We continue with our running example, illustrated in Figure 6. Our goal is to ensure that the $+\phi$ feature on T is paired with a $-\phi$ feature on the closest visible DP in its c-command domain, which is normally the verbal subject. To do this, we extract the (main) spinal c-string and project a tier which includes all elements which are potential bearers of $\pm\phi$, that is, all D heads and finite T heads. In addition, we project C since agreement cannot cross a finite or non-finite CP boundary, nor can it skip a CP subject and agree with a DP object (this will be discussed in detail momentarily). On the tier, we require that every element bearing $+\phi$ be immediately followed by one bearing $-\phi$, and that every element bearing $-\phi$ be immediately preceded by one bearing $+\phi$. In

[7] Following common practice, I abbreviate the bundle of ϕ-features as a single feature when they act together, as is true of the examples in this paper.

[8] An exploration of alternative feature systems, such as the four-way split in Pesetsky and Torrego 2007, is beyond the scope of this paper. Upon initial consideration, it seems unlikely that there will be any major formal differences. In fact, as a reviewer notes, just a single undifferentiated ϕ diacritic would be sufficient for the patterns studied here.

the current structure, this constraint is satisfied: our tier consists of T followed by two D heads, and the only probe and goal are adjacent.

Next, we consider some ways in which this constraint could be violated, which correspond to the banned substrings in the TSL grammar. For comparison, the licit example from Figure 6 is repeated in (4a). (For simplicity, I swap out most lexical items for their categories when writing out c-strings and their tiers, and also omit movement features when not relevant.) First, it is not possible for T to agree with any DP other than the closest. For example, it cannot bypass the subject in favor of agreement with the object, as in (4b). In such a structure, the subject D head intervenes on the tier, violating both clauses of our constraint. Formally, we say that the tier substrings $\mathrm{T}_{[+\phi]} \cdot \mathrm{D}$ and $\mathrm{D} \cdot \mathrm{D}_{[-\phi]}$ are illicit. Similarly, agreement with the object in (4c) is impossible because the subject C head intervenes, even though by hypothesis it cannot agree. Finally, agreement into CP is impossible (4d), though agreement into non-finite TP is possible; see (6) below.[9]

(4) Licit and illicit subject-verb agreement configurations

a. Subject agreement (✓)
  - Sentence: The cat **chases** the rats.
  - C-string: $\mathrm{T}_{[+\phi]} \cdot v \cdot \mathrm{D}_{[-\phi]} \cdot \mathrm{V} \cdot \mathrm{D} \cdot \mathrm{N}$
  - ϕ-agreement tier: $\mathrm{T}_{[+\phi]} \cdot \mathrm{D}_{[-\phi]} \cdot \mathrm{D}$
  - Constraints violated: n/a

b. Object agreement across DP (✗)
  - Sentence: The cat **chase** the rats.
  - C-string: $\mathrm{T}_{[+\phi]} \cdot v \cdot \mathrm{D} \cdot \mathrm{V} \cdot \mathrm{D}_{[-\phi]} \cdot \mathrm{N}$
  - ϕ-agreement tier: $\mathrm{T}_{[+\phi]} \cdot \mathrm{D} \cdot \mathrm{D}_{[-\phi]}$
  - Constraints violated: $\mathrm{T}_{[+\phi]} \cdot \mathrm{D}$, $\mathrm{D} \cdot \mathrm{D}_{[-\phi]}$

c. Object agreement across CP (✗)
  - Sentence: [$_{\mathrm{CP}}$ That he plays the bassoon] **impress** us.
  - C-string: $\mathrm{T}_{[+\phi]} \cdot v \cdot \mathrm{C} \cdot \mathrm{V} \cdot \mathrm{D}_{[-\phi]}$
  - ϕ-agreement tier: $\mathrm{T}_{[+\phi]} \cdot \mathrm{C} \cdot \mathrm{D}_{[-\phi]} \cdot$
  - Constraints violated: $\mathrm{T}_{[+\phi]} \cdot \mathrm{C}$, $\mathrm{C} \cdot \mathrm{D}_{[-\phi]}$

[9] I use a non-finite embedded clause in this example to avoid a confound with finite clauses, which is that finite embedded T intervenes even if C is invisible.

d. Agreement into non-finite CP (✗)
   - Sentence: It **are** possible [$_{CP}$ for rats to have fleas.]
   - C-string: $T_{[+\phi]} \cdot \text{be} \cdot A \cdot C \cdot \text{to} \cdot v \cdot D_{[-\phi]} \cdot V \cdot D \cdot N$
   - ϕ-agreement tier: $T_{[+\phi]} \cdot C \cdot D_{[-\phi]} \cdot D$
   - Constraints violated: $T_{[+\phi]} \cdot C$, $C \cdot D_{[-\phi]}$

Some readers may be wondering about the treatment of C as an arbitrary blocker. Admittedly, English is not an ideal example of this since there are alternative analyses available in most cases. For example, we could posit that CPs do agree, but that this agreement is always singular. At the same time, it is descriptively true that ϕ-agreement is blocked whenever a CP intervenes. This aligns with the behavior of EPP movement, but contrasts sharply with *wh*-movement, in which declarative C does not interfere. As discussed by Keine (2019), opaqueness of a given type of phrase must be relativized to individual dependencies, even in theories which include successively cyclic movement and/or phases. The blockers for specific dependencies such as ϕ-agreement also vary across languages. This issue will be explored further in Section 4.1.

Two remaining ways our constraint could be violated include tiers which contain two probes or two goals in sequence, that is, those that contain substrings such as $T_{[+\phi]} \cdot T_{[+\phi]}$ or $D_{[-\phi]} \cdot D_{[-\phi]}$.[10] Putting all of this together, we arrive at the (informal) TSL-2 grammar shown in (5) below. Here, I introduce several additional notational shortcuts. We are already using T/D/C as a stand-in for any item of the relevant category; in addition, X will be used as a placeholder for an element of any category. Next, when a category is followed by a list of features in square brackets, this denotes an element bearing *exactly* those features, while X with no brackets denotes an element with no relevant features (in this case, $\pm\phi$). It is understood that the tier alphabet should be compiled out to all of the matching symbols in the MG lexicon, and likewise for the banned substrings on the tier. Note that the alphabet Σ contains all elements in the MG lexicon; this never varies for a given language, so I omit it from all grammar definitions.

[10] It is difficult to think of a context in English where we could find two probes not separated by a CP boundary, but since any DP can bear $-\phi$, a sequence of two goals could potentially occur in any transitive clause.

(5) ϕ-agreement tier for English

- Project: all finite T, all D, all C
- Constraints: $\left\{ \begin{array}{cc} X_{[+\phi]} \cdot X_{[+\phi]}, & X_{[-\phi]} \cdot X_{[-\phi]}, \\ X_{[+\phi]} \cdot X, & X \cdot X_{[-\phi]}, \\ X_{[+\phi]} \cdot \ltimes, & \rtimes \cdot X_{[-\phi]} \end{array} \right\}$

The overall analysis is extremely similar to the analysis of movement in Graf 2022b. The primary difference is that because we are using c-strings, we are able to handle relativized minimality in full generality; with tree tiers, only blocking by containing elements can be handled correctly. Another notable characteristic of the model is that domain-based and intervention-based blocking are treated uniformly, as exemplified by our treatment of intervention by C.

These, of course, are the simple cases; even in English, there are situations when the correlation between subject-hood and agreement comes apart. This happens in existential sentences like (6). For whatever reason, existential *there* seems to be invisible for agreement. Assuming it to be absent from the ϕ-agreement tier, long-distance agreement with the embedded subject follows, as the latter is adjacent on the tier just like a canonical subject.[11] In addition, long-distance agreement across *there* is optional for many, if not most English speakers, an issue which we will revisit in Section 4.4.

(6) Long-distance agreement in existential sentences

- Sentence: There seem [$_{\text{TP}}$ to be [$_{\text{PredP}}$ some squirrels in the attic]].
- C-string: $\text{T}_{[+\text{EPP},+\phi]} \cdot v \cdot \text{seem} \cdot \text{to} \cdot \text{be} \cdot \text{there}_{[-\text{EPP}]} \cdot \text{Pred} \cdot \text{D}_{[-\phi]} \cdot \text{P} \cdot \text{D} \cdot \text{N}$
- ϕ-agreement tier: $\text{T}_{[+\phi]} \cdot \text{D}_{[-\phi]} \cdot \text{D}$
- Constraints violated: n/a

At this point, I should mention an alternative model, which is to posit that all lexical items enter the derivation with concrete feature values (1SG/1PL/etc.), and the TSL grammar checks that they match

[11] I assume that *there* originates in Spec-*v*P of specific verbs including *be*; see Deal 2009 for arguments in favor of this analysis. I also assume that the complement of *be* is a PredP, though nothing crucial hinges on this.

in configurations where agreement applies. In this case, the tier constraints for subject-verb agreement would require finite T to bear the same set of valued ϕ-features as the following D node. This is analogous to the treatment of phonological harmony in Section 2.3, and is also similar to the checking model of agreement in early Minimalism. I believe there is value in such an approach, but the present system more clearly highlights the structural configurations of the agreeing heads, which are the primary focus of this paper.

There is also an existing version of MG which handles agreement (Ermolaeva 2018; Ermolaeva and Kobele 2022, 2023). In this system, agreement occurs via dependencies created by Merge and Move, and is restricted through subdiacritics on the relevant MG features on nodes along dependency paths. This is analogous to using c-strings obtained from multidominance trees, an intriguing possibility which merits future exploration. One disadvantage of the model is that long-distance agreement requires either covert movement or passing of features along unbounded selectional chains in the absence of any morphological realization. The former, assuming covert movement to affect scope, contradicts recent empirical findings, including the famous Tsez data (Polinsky and Potsdam 2001). The latter is problematic from a subregular perspective, since arbitrarily complex selectional features can simulate any regular tree constraint (cf. Rogers 1997; Graf 2013).

There are yet other reasons to assume agreement to be independent from movement. First and foremost, it would be be methodologically backwards to do otherwise, given that we are trying to establish their formal properties in the first place. Also, even if the claim that both phenomena are TSL-2 is upheld, this does not imply that they must be unified in the grammar; instead, each can be seen as an independent manifestation of the same underlying computational resources (cf. Graf 2022a). Likewise, I assume case to be assigned/licensed independently; see Hanson 2023 for a subregular approach to case that uses MG dependency trees. In summary, I treat agreement, movement, and case dependencies as being essentially autonomous, though they may interact when one tier grammar makes reference to features that are themselves regulated by another tier. Multiple examples of this sort appear in the following section, in which I survey a wide variety of agreement patterns from the syntactic literature and show that they are all TSL-2.

4 THE TYPOLOGY OF AGREEMENT

Graf (2022b) showed how the space of parameters made available by a TSL-2 grammar closely matches the attested variation in movement patterns across languages. Here, I do the same for ϕ-agreement. We begin with examples of variation in the set of elements which are projected on the tier, which together with their features controls the set of agreeing, invisible, and blocking elements (Section 4.1). In addition, TSL-2 also permits variation in directionality (Section 4.2), as well as seemingly complex configurations in which multiple probes share a single goal (Section 4.3) or a single probe interacts with multiple goals (Section 4.4). The section closes by revisiting the power of TSL-2 and its alignment with the observed typology (Section 4.5).

An overview of these parameters of the grammar and the corresponding agreement patterns is given in Table 1; the full set of patterns treated in this paper is summarized in Table 2 at the end of this section. The existence of such patterns is hardly a mystery, but rather to be expected if agreement is TSL-2. We do not expect every logically possible pattern to be attested since, as discussed briefly in the introduction, there are other factors influencing typology, including but not limited to constraints on acquisition and diachronic development. But we do expect to find a reasonably diverse subset of the patterns made possible by the computational system, and this is certainly the case for agreement.

Table 1: Variants of a TSL-2 grammar and corresponding agreement phenomena

	Tier projection	Tier constraints	Phenomenon
a.	All $\pm\phi$ elements	Strict matching of $+\phi/-\phi$	Minimality
b.	Some D heads do not project	(as in (a))	Invisibility
c.	Some non-agreeing items also project	(as in (a))	Blocking
d.	(as in (a))	Swap order of $+\phi/-\phi$	Upward agreement
e.	(as in (a))	Allow sequential $+\phi$	Chain agreement
f.	(as in (a))	Allow sequential $-\phi$	Multiple agreement

It is not possible to conduct an exhaustive survey here, so I have chosen to focus on two major themes – case-sensitive agreement and complementizer agreement – in order to show both that the same formal patterns occur across agreement phenomena and that various patterns are attested within a single phenomenon. In particular, I do not treat any patterns in which subfeatures of ϕ act independently; while clearly important, in principle these involve multiple tiers and are therefore beyond the scope of this paper. Similarly, I do not provide a detailed analysis of patterns in which a single predicate overtly agrees with multiple DPs, for reasons discussed in Section 2.4. However, I will sketch how this could be done in terms of the interaction-satisfaction theory (Deal 2015), applying the approach to the optionality problem mentioned above and discussing some of the caveats.

Invisibility and blocking 4.1

We have already seen both invisibility and blocking in action, even in the simple examples of English subject-verb agreement: elements such as V and *there* are invisible while D heads block agreement with more distant DPs, as do C heads. The visibility conditions for EPP movement are similar, but declarative C does not block *wh*-movement, and so does not appear on the corresponding tier. Furthermore, exactly which elements agree, block agreement, or are invisible varies across languages. In this section, we examine several examples of case-sensitive agreement, in which the behavior of a DP depends on its case.

Our first example comes from Hindi, which features split ergative case marking conditioned by aspect (Mahajan 1990). Imperfective clauses have a nominative-accusative pattern while perfective clauses are ergative-absolutive, as shown in (7).[12] I assume that nominative

[12] Abbreviations used in this paper: 1 = first person, 2 = second person, 3 = third person, C1 = class 1, C2 = class 2, CS = construct state, DAT = dative, DEM = demonstrative, ERG = ergative, F = feminine, FUT = future, GEN = genitive, HAB = habitual, IPFV = imperfective, LOC = locative, M = masculine, NF = non-finite, NOM = nominative, OV = object voice, PASS = passive, PFV = perfective, PL = plural, PRF = perfect, PST = past, SG = singular, SV = subject voice.

and absolutive case (both unmarked) are the same case, call it nominative. Also, Hindi displays differential object marking, so not all objects are overtly case-marked; I gloss these as nominative as well. We see that in the imperfective clause the verb agrees with the subject, while in the perfective clause it agrees with the object. Thus, descriptively, the verb agrees with the highest nominative DP, and ergative DPs are invisible.

(7) Case-sensitive agreement in Hindi (Mahajan 1990)

a. Raam roTii khaataa thaa.
Raam.M.**NOM** bread.F.NOM eat.IPFV.M be.PST.M
'Raam ate bread (habitually).'

b. Raam-ne roTii khaayii.
Raam.M-**ERG** bread.F.NOM eat.PFV.F
'Raam ate bread.'

As discussed in the previous section, I assume that case information is available in the form of syntactic features like those for movement and agreement; see Preminger 2014 for a syntactic argument in favor of this idea. I also assume that agreement is conditioned on case-marking rather than the reverse (Bobaljik 2008). Thus, all we need to do to capture the Hindi agreement pattern is to modify the tier alphabet for the ϕ-agreement tier: instead of projecting all DPs, we project only nominative DPs, since only these are ever eligible for agreement. The tier constraints remain unchanged.

Let us confirm that this analysis derives the correct results. I assume the same basic T/*v*/V clause structure as in English unless there are relevant differences. In the imperfective Hindi clause, we have an additional auxiliary verb, which can be assumed to occupy an Asp(ect) projection. The resulting structures for the sentences in (7) are as shown in Figure 7. For simplicity, I will ignore the agreement on the non-finite verb, focusing just on the finite verb (we will return to the issue of multiple agreeing elements in Section 4.3).

C-strings and their ϕ-agreement tiers for these examples are shown in (8) along with the illicit opposite agreement configurations. In the imperfective clause both subject and object are projected, so only the subject can agree, as in (8a); if $-\phi$ is placed on the object (8b), the subject intervenes, resulting in a minimality violation. In the

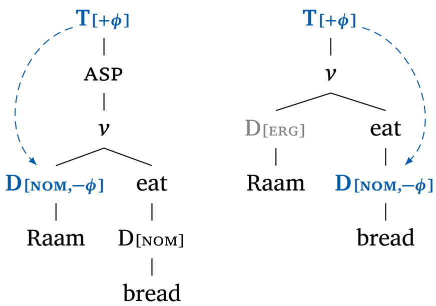


Figure 7:
Case-sensitive agreement in Hindi. Ergative DPs are invisible, causing the object to agree instead of the subject

imperfective clause, this is reversed. If $-\phi$ is placed on the subject, this will not work as it is ergative and therefore does not appear on the tier, leaving a lone probe (8c). Instead, it is the object that must agree, since it is adjacent to T on the tier (8d). This is also what happened in the example of agreement across existential *there* in Section 3.4.

(8) Example c-strings and tier projections for Hindi

a. Imperfective clause, subject agreement (✓)
   - C-string: $\mathrm{T}_{[+\phi]} \cdot \mathrm{ASP} \cdot v \cdot \mathrm{D}_{[\textsc{nom},-\phi]} \cdot \mathrm{V} \cdot \mathrm{D}_{[\textsc{nom}]} \cdot \mathrm{N}$
   - ϕ-agr. tier: $\mathrm{T}_{[+\phi]} \cdot \mathrm{D}_{[\textsc{nom},-\phi]} \cdot \mathrm{D}_{[\textsc{nom}]}$
   - Constraints violated: n/a

b. Imperfective clause, object agreement (✗)
   - C-string: $\mathrm{T}_{[+\phi]} \cdot \mathrm{ASP} \cdot v \cdot \mathrm{D}_{[\textsc{nom}]} \cdot \mathrm{V} \cdot \mathrm{D}_{[\textsc{nom},-\phi]} \cdot \mathrm{N}$
   - ϕ-agr. tier: $\mathrm{T}_{[+\phi]} \cdot \mathrm{D}_{[\textsc{nom}]} \cdot \mathrm{D}_{[\textsc{nom},-\phi]}$
   - Constraints violated: $\mathrm{X}_{[+\phi]} \cdot \mathrm{X}$, $\mathrm{X} \cdot \mathrm{X}_{[-\phi]}$

c. Perfective clause, subject agreement (✗)
   - C-string: $\mathrm{T}_{[+\phi]} \cdot v \cdot \mathrm{D}_{[\textsc{erg},-\phi]} \cdot \mathrm{V} \cdot \mathrm{D}_{[\textsc{nom}]} \cdot \mathrm{N}$
   - ϕ-agr. tier: $\mathrm{T}_{[+\phi]} \cdot \mathrm{D}_{[\textsc{nom}]}$
   - Constraints violated: $\mathrm{X}_{[+\phi]} \cdot \mathrm{X}$

d. Perfective clause, object agreement (✓)
   - C-string: $\mathrm{T}_{[+\phi]} \cdot v \cdot \mathrm{D}_{[\textsc{erg}]} \cdot \mathrm{V} \cdot \mathrm{D}_{[\textsc{nom},-\phi]} \cdot \mathrm{N}$
   - ϕ-agr. tier: $\mathrm{T}_{[+\phi]} \cdot \mathrm{D}_{[\textsc{nom},-\phi]}$
   - Constraints violated: n/a

It should be noted that some Hindi verbs take dative subjects, and that these are also invisible.[13] In other words, it really is only

[13] The same is true of marked objects, which can be considered to bear accusative case. Thus, when the subject is either ergative or dative and the object

nominatives that can agree. To summarize, the TSL grammar for Hindi is shown in (9). From this point forward, I will highlight what has changed in comparison to the English grammar from Section 3.4. In this case, only the tier projection rules have changed.[14]

(9) ϕ-agreement tier for Hindi

- Project: all T, **D if [NOM]**, all C
- Constraints: $\left\{\begin{array}{cc} X_{[+\phi]} \cdot X_{[+\phi]}, & X_{[-\phi]} \cdot X_{[-\phi]}, \\ X_{[+\phi]} \cdot X, & X \cdot X_{[-\phi]}, \\ X_{[+\phi]} \cdot \ltimes, & \rtimes \cdot X_{[-\phi]} \end{array}\right\}$

It should also be emphasized that there is nothing inherent about the (in)visibility of certain cases. In Nepali, which is closely related to Hindi, the verb agrees with the subject whether it is ergative or nominative (Coon and Parker 2019), as shown in (10).

(10) Case-insensitive agreement in Nepali (Coon and Parker 2019)

a. Maile yas pasal-mā patrikaā kin-ē.
1SG.ERG DEM store-LOC newspaper.NOM buy-**1SG**
'I bought the newspaper in this store.'

b. Ma thag-ī-ē.
1SG.NOM cheat-PASS-**1SG**
'I was cheated.'

Nepali also allows dative subjects, and these do not agree, just as they do not in Hindi. Broadly speaking, there appears to be a hierarchy for case visibility in which unmarked case (nominative) ranks above dependent cases (accusative and ergative), followed by oblique cases (dative); each language chooses a point along the hierarchy below which DPs are invisible for agreement (Bobaljik 2008). As such, Nepali is best characterized as a language in which DPs bearing unmarked or

is accusative, there is no DP which is eligible for agreement. In this case, default agreement arises. We will see an example of this momentarily in our discussion of Icelandic, so I omit treatment of this phenomenon here.

[14] A reviewer expressed concern about the potential power of these conditional tier projection rules. Because the lexicon is finite, so too is the set of possible tier projections. Therefore, we gain no power compared to exhaustively listing every item. The current notation serves only as a convenient shorthand.

dependent case are visible, but those with an oblique case are not. We can easily encode such information in our TSL grammar. In the case of Nepali agreement, we simply project DPs if they are nominative or ergative, though not if they are dative, as in (11).

(11) ϕ-agreement tier for Nepali

- Project: all T, **D if [NOM/ERG]**, all C
- Constraints: $\left\{\begin{array}{cc} X_{[+\phi]} \cdot X_{[+\phi]}, & X_{[-\phi]} \cdot X_{[-\phi]}, \\ X_{[+\phi]} \cdot X, & X \cdot X_{[-\phi]}, \\ X_{[+\phi]} \cdot \ltimes, & \rtimes \cdot X_{[-\phi]} \end{array}\right\}$

Of course, we could do this just as easily for a combination of cases which violates the case visibility hierarchy. However, there is good reason to think that implicational hierarchies such as this one should be attributed to extragrammatical factors, since they are just one member of a much larger class of monotonicity effects which are prevalent throughout language; see Graf 2020 for discussion.[15]

Next, let us turn our attention to blocking effects. What is especially interesting about dative DPs is that while they are often invisible for agreement, they are also known to block it, a phenomenon known as *dative intervention*. A famous example comes from Icelandic, where both possibilities occur depending on the structure. First, (12) is an example of agreement across a dative subject in a simple transitive clause, demonstrating invisibility; next, (13) shows the transitive expletive construction, one of the contexts where datives instead block agreement. It is important to note that the singular verb form in (13a) is not agreement with the dative DP, but a default ending.

(12) Dative invisibility in Icelandic
(Holmberg and Hróarsdóttir 2003)

Henni	likuðu	hestarnir.
her.**SG**.**DAT**	liked.**PL**	the.horse.**PL**.NOM

'She liked the horses.'

[15] Note that even if the grammar formalism is formulated in such a way so as to enforce such hierarchies, this still does not explain their existence, but rather raises the question as to why the grammar should be this way.

(13) Dative intervention in transitive expletives
(Holmberg and Hróarsdóttir 2003)

a. Það finnst einhverjum stúdent
EXPL find.**SG** some student.**SG**.**DAT**
[tölvurnar ljótar].
the.computer.**PL**.NOM ugly

b. * Það finnast einhverjum stúdent
EXPL find.**PL** some student.**SG**.**DAT**
[tölvurnar ljótar].
the.computer.**PL**.NOM ugly
'Some student finds the computers ugly.'

It is well known that non-nominative subjects do not trigger subject-verb agreement in Icelandic (Andrews 1982; Thrainsson 2007, a.o.). This applies not only to dative subjects, but also to genitive and accusative subjects when they occur. In most cases, the result is long-distance agreement with a lower nominative, as in Hindi. But, in the transitive expletive construction, dative subjects intervene, at least as a first approximation. The full data is quite complex, as the visibility of a dative DP is determined in part by whether it undergoes a specific type of movement which is not always available; on top of this, long-distance agreement is subject to dialectal differences and is optional for certain speakers under certain conditions (Sigurðsson and Holmberg 2008; Kučerová 2016). For now, let us focus just on the above data, in which datives are blockers. Later, we will deal with the interaction with movement. I do not attempt to treat optionality here, though we will consider several approaches to optionality in Section 4.4 which could potentially be applied to the Icelandic data.

I assume the transitive expletive construction to involve a small clause structure, modeled here as a PredP, as shown in Figure 8. Since dative DPs block agreement with a more distant DP, they must be projected on the ϕ-tier just like nominatives. As for the fact that we get default agreement in cases of dative intervention, there are several plausible ways in which this could be analyzed, each of which is TSL. For concreteness, let us assume the default agreement means that T has not agreed with anything. In other words, we can have a

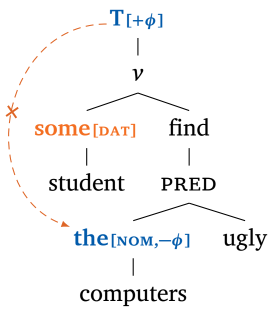


Figure 8:
Dative intervention in Icelandic transitive expletive construction. Dative DPs do not agree, but also block agreement with a more distant DP

lone probe without a goal, at least in this specific circumstance.[16] We modify the constraints accordingly, banning a probe which is immediately followed by a non-agreeing DP only when it is nominative and therefore eligible for agreement.

The provisional grammar for Icelandic is given in (14), with the relevant constraint modification highlighted. Next, c-strings and their tiers for the licit and illicit agreement configurations in (13) are shown in (15). This time, the items that are projected are the same in either case; what differs is whether the lower nominative bears $-\phi$. When it does, the derivation is illicit since it is preceded by the higher dative. When it does not, no constraints are violated, since T need not agree in this context.

(14) ϕ-agreement tier for Icelandic (provisional)

- Project: all T, all D, all C
- Constraints: $\left\{ \begin{array}{cc} X_{[+\phi]} \cdot X_{[+\phi]}, & X_{[-\phi]} \cdot X_{[-\phi]}, \\ X_{[+\phi]} \cdot \mathbf{D}_{[\textsc{nom}]}, & X \cdot X_{[-\phi]}, \\ X_{[+\phi]} \cdot \ltimes, & \rtimes \cdot X_{[-\phi]} \end{array} \right\}$

(15) C-strings and tier projections for Icelandic transitive expletives

a. Default agreement (✓)

- C-string: $T_{[+\phi]} \cdot v \cdot D_{[\textsc{dat}]} \cdot V \cdot \text{Pred} \cdot D_{[\textsc{nom}]} \cdot A$
- ϕ-agr. tier: $T_{[+\phi]} \cdot D_{[\textsc{dat}]} \cdot D_{[\textsc{nom}]}$
- Constraints violated: n/a

[16] In Preminger's (2014) terms, agreement is an obligatory operation in the sense that it must occur when applicable, not that it must occur no matter what.

b. Agreement across dative DP (✗)

- C-string: $T_{[+\phi]} \cdot v \cdot D_{[\text{DAT}]} \cdot V \cdot \text{Pred} \cdot D_{[\text{NOM},-\phi]} \cdot A$
- ϕ-agr. tier: $T_{[+\phi]} \cdot D_{[\text{DAT}]} \cdot D_{[\text{NOM},-\phi]}$
- Constraints violated: $X \cdot X_{[-\phi]}$

In the above analysis, we are assuming that the dative DP cannot itself bear $-\phi$; in contrast, finite T is lexically specified to always carry $+\phi$ since it must agree when possible. A common alternative is to say that datives do agree, but that default features are transmitted. In this analysis, the dative subject does bear $-\phi$ in the licit agreement configuration, and agreement with the lower nominative is just an ordinary minimality violation. This situation, where the fine details of the analysis have no bearing on whether the phenomenon in question is TSL, seems to be quite common, and points to the robustness of the computational characterization of the empirical facts; we will encounter several more examples like this later in this paper.

At this point, we have seen how our grammar can be adjusted to account for DPs which are invisible to or block agreement according to their case. We can also handle variable visibility within a single language, which as mentioned above is a core aspect of the Icelandic pattern. Again, the full data is notoriously complex, so to keep the discussion simple while still addressing the relevant computational issue, we will add just one additional data point. Recall that dative subjects in simple transitive clauses are invisible. Long-distance agreement is also possible in sentences analogous to the transitive expletive construction, but in which no expletive is inserted and the logical subject raises to Spec-TP, as is assumed in simplex sentences like (12). This is shown in (16).

(16) Long-distance agreement when dative DP moves to Spec-TP (Holmberg and Hróarsdóttir 2003)

Einhverjum	stúdent	finnast	[tölvurnar
some	student.**SG.DAT**	find.**PL**	the.computer.**PL**.NOM
ljótar].			
ugly			

'Some student finds the computers ugly.'

Based on this data, we would say that datives are invisible precisely when they undergo EPP movement.[17] Accordingly, we project all nominatives, plus datives which do *not* bear −EPP; other non-nominative DPs are always invisible for agreement. Since default agreement also occurs in intransitive sentences with a dative subject, such as (12), whose tier would contain $T_{[-\phi]}$ at the right edge of the tier, we also remove the constraint $X_{[+\phi]} \cdot \ltimes$. The revised grammar is presented in (17) below. The tier projection rule for D heads is fairly complex; I have taken what I believe to be the least confusing option, which is to list nominative and dative DPs separately. Similarly, I indicate that a constraint has been removed by striking it out.

(17) ϕ-agreement tier for Icelandic (revised)

- Project: all T, **all $D_{[\textsc{nom}]}$**, **$D_{[\textsc{dat}]}$ if not [−EPP]**, all C
- Constraints: $\left\{ \begin{array}{cc} X_{[+\phi]} \cdot X_{[+\phi]}, & X_{[-\phi]} \cdot X_{[-\phi]}, \\ X_{[+\phi]} \cdot \mathbf{D_{[\textsc{nom}]}}, & X \cdot X_{[-\phi]}, \\ \cancel{X_{[+\phi]} \cdot \ltimes}, & \rtimes \cdot X_{[-\phi]} \end{array} \right\}$

The situation with visibility of dative DPs in Icelandic is essentially the opposite of the differential object marking pattern mentioned in Section 3.4, in which the DP becomes visible to its case assigner if and only if it *does* move. Either way, we must refer to a movement feature on the ϕ-tier just as we did with case in Hindi. As I will argue in the following sections, there are yet other movement features which interact with agreement in a similar way.

4.2 *Directionality*

The issue of directionality has received considerable attention in the theoretical literature on agreement. Theories differ as to whether it is always downward (Chomsky 2000), always upward (Zeijlstra 2012), or varies parametrically (Baker 2008). (To be clear, when we say here that agreement is downward, this means that the goal appears below

[17] According to Kučerová (2016), the correct generalization is that dative DPs in Icelandic are invisible if they undergo object shift (movement to Spec-*v*P), and intervene if they do not. This movement tends to be unavailable in the transitive expletive construction for semantic reasons.

the probe, and vice versa for upward agreement.) Despite this theoretical disagreement, from the present perspective agreement is clearly predicted to be able to proceed in either direction. This is because TSL patterns do not have any fixed notion of directionality: for any two elements X and Y, the grammar may allow XY, YX, both, or neither. Thus, it is not surprising that in phonology we find both progressive and regressive harmony. Likewise, in syntax subject-verb agreement is usually downward looking (since the subject is below T in the derivation tree), while negative concord is upward looking, as is case concord within the DP (if case is inherent on D, as we have been assuming).[18]

Ideally, we would like to see evidence of variation within a single agreement phenomenon. This appears at first glance to be true of complementizer agreement (Diercks 2013). In West Flemish, we find cases where the complementizer heading an embedded clause agrees downward for number with the embedded subject (18). In contrast, in Lubukusu (a Bantu language spoken in Kenya), we find upward agreement for noun class with the next higher subject (19). Note that not all complementizers agree in Lubukusu.

(18) Downward agreement in West Flemish (Diercks 2013)

a. Kpeinzen da-j [(gie) morgen goat].
I.think that-**you** **(you)** tomorrow go
'I think that you'll go tomorrow.'

b. Kvinden dan [die boeken te diere zyn].
I.find that.**PL** the book.**PL** too expensive be.**PL**
'I find those books too expensive.'

(19) Upward agreement in Lubukusu (Diercks 2013)

a. Ba-ba-ndu ba-bolela Alfredi [ba-li a-kha-khile].
C2-**C2**-people **C2**-said C1.Alfred **C2**-that C1-FUT-conquer
'The people told Alfred that he will win.'

[18] Even if a harmony pattern is symmetric on the surface, as in the example in Section 2.3, the process that generates it may be clearly directional. TSL string languages have been generalized to functions to model such processes; see Burness *et al.* 2021 for an overview.

b. Alfredi ka-bolela ba-ba-ndu [a-li
C1.Alfred **C1**-said C2-C2-people **C1**-that
ba-kha-khile].
C2-FUT-conquer
'Alfred told the people that they will win.'

This data is particularly informative in that in neither case does the agreeing DP move out of the embedded CP, avoiding ambiguity in directionality of agreement depending on whether the DP agrees from the lower or higher position. However, there is a complication. In Diercks' analysis of Lubukusu, the matrix clause subject actually binds an operator in embedded Spec-CP, which in turn agrees locally with embedded C, a process which he calls *indirect agreement*. In our terms, this means that the (local) agreement between C and the operator is downward, though the binding relation between the operator and the subject is still upward in the sense that the bound operator must be licensed from above. For comparison, Figure 9 shows Diercks' analysis for (19a) alongside the direct agreement analysis.

Diercks' arguments against upward agreement in Lubukusu can be summarized as follows: 1) agreement is strictly subject-oriented, and other intervening DPs are ignored; 2) in subject questions the verb follows a reduced agreement paradigm while complementizer agreement is as usual; and 3) Lubukusu also features hyperraising (i.e. raising out of a finite clause), and complementizer agreement is absent when this occurs across a C head which is otherwise expected to agree. None of these arguments hold up. First, regarding subject-orientation, we have

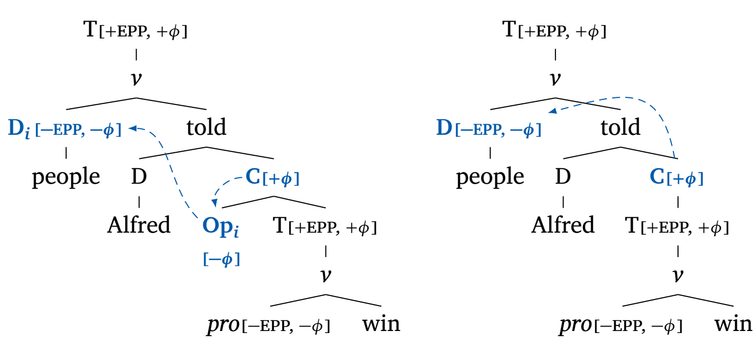


Figure 9: Complementizer agreement in Lubukusu. Left: indirect analysis. Right: direct agreement. In either analysis, one upward dependency is involved

already seen multiple examples where agreement targets only a subset of DPs (e.g. depending on case) and the literature contains many besides these. We can easily pick out the subject in the TSL analysis as the DP which undergoes EPP movement. Second, the fact that subject-verb agreement may sometimes follow a different paradigm is irrelevant, as there is no reason to think that complementizer agreement here is parasitic on verbal agreement. Finally, the lack of complementizer agreement in hyperraising constructions is in fact predicted by the present model, since the subject appears only below C in the dependency tree and is thus invisible to upward looking dependencies.[19]

While Diercks presents additional arguments in favor of his indirect agreement analysis, they are circumstantial at best. For example, he draws a parallel between certain blockers of complementizer agreement and well-known binding phenomena, which is suggestive of the presence of a bound variable. But as we have already discussed, blocking conditions for agreement can also be quite complex and varied, and there is no particular reason to think that this data can only be explained in terms of binding. This being the case, it is simpler to dispense with the bound operator and assume direct agreement.

Before continuing, I wish to stress that whether we choose to analyze the dependency in question as binding or agreement, the formal shape of the pattern is identical; the only difference is whether the lower element is the C head itself or an operator in its specifier. Furthermore, even if Diercks' analysis is correct for Lubukusu, this would not be enough to discount the existence of upward agreement as a whole; indeed, he does not discuss any other possible instances.

That said, there are in principle several ways to implement variable directionality, and I see no strong reason to prefer one over the others, so I will take the obvious route and simply specify that agreeing C must follow its goal on the tier rather than precede it. (An alternative will be discussed at the end of this section.) Also, since the agreement

[19] This last fact is somewhat mysterious in a Minimalist analysis, since without additional assumptions there is no reason why a raised subject should not value the lower complementizer, or alternatively bind the operator in Spec-CP. Diercks (2013) proposes one possible solution. Under the present model, no special treatment is required. In contrast, the present model may struggle when movement feeds subsequent operations, as mentioned in Section 3.2.

on C is strictly subject-oriented, only DPs bearing −EPP are projected on the ϕ-agreement tier, as in our treatment of case-sensitive agreement. Finally, for simplicity I will ignore agreement with T, assuming it to be regulated on a separate tier as discussed in Section 2.4.

As usual, a selection of licit and illicit c-strings are given in (20), representing variants of example (19a). First, we have agreement with the upper subject, shown in (20a). The intervening object is invisible, so the subject immediately precedes the agreeing C head, as required. Next are two illicit configurations, in which agreement is attempted with the upper object (20b) and the lower subject (20c), respectively. The former does not work because the object is missing from the tier, leaving the probe on C without a goal. In the latter case, the goal follows the probe rather than preceding it, violating multiple constraints.

(20) C-strings and tier projections for Lubukusu

a. Complementizer agrees with upper subject (✓)
   - C-string: $\mathrm{T}_{[+\text{EPP}]} \cdot v \cdot \mathbf{D}_{[-\text{EPP},\,-\phi]} \cdot \mathrm{V} \cdot \mathrm{D} \cdot \mathbf{C}_{[+\phi]} \cdot \mathrm{T}_{[+\text{EPP}]} \cdot v \cdot \mathrm{D}_{[-\text{EPP}]} \cdot \mathrm{V}$
   - Tier: $\mathbf{D}_{[-\text{EPP},\,-\phi]} \cdot \mathbf{C}_{[+\phi]} \cdot \mathrm{D}_{[-\text{EPP}]}$
   - Constraints violated: n/a

b. Complementizer agrees with upper object (✗)
   - C-string: $\mathrm{T}_{[+\text{EPP},\,+\phi]} \cdot v \cdot \mathrm{D}_{[-\text{EPP}]} \cdot \mathrm{V} \cdot \mathbf{D}_{[-\phi]} \cdot \mathbf{C}_{[+\phi]} \cdot \mathrm{T}_{[+\text{EPP},\,+\phi]} \cdot v \cdot \mathrm{D}_{[-\text{EPP}]} \cdot \mathrm{V}$
   - Tier: $\mathrm{D}_{[-\text{EPP}]} \cdot \mathbf{C}_{[+\phi]} \cdot \mathrm{D}_{[-\text{EPP}]}$
   - Constraints violated: $\mathrm{X} \cdot \mathrm{C}_{[+\phi]}$

c. Complementizer agrees with lower subject (✗)
   - C-string: $\mathrm{T}_{[+\text{EPP}]} \cdot v \cdot \mathrm{D}_{[-\text{EPP}]} \cdot \mathrm{V} \cdot \mathrm{D} \cdot \mathbf{C}_{[+\phi]} \cdot \mathrm{T}_{[+\text{EPP}]} \cdot v \cdot \mathbf{D}_{[-\text{EPP},\,-\phi]} \cdot \mathrm{V}$
   - Tier: $\mathrm{D}_{[-\text{EPP}]} \cdot \mathbf{C}_{[+\phi]} \cdot \mathbf{D}_{[-\text{EPP},\,-\phi]}$
   - Constraints violations: $\mathrm{X} \cdot \mathrm{C}_{[+\phi]}$, $\mathrm{X}_{[-\phi]} \cdot \ltimes$

There is a potential problem with this analysis: if $-\phi$ is placed on both the subject and the object, then we get the same result as if it appears only on the subject, but only the latter configuration should be licit. This can be avoided by specifying in the lexicon that only D heads with −EPP may also bear $-\phi$. Similarly, though we have not discussed any examples of non-agreeing complementizers, these are distinguished from agreeing complementizers in that the former never bear $-\phi$, while the latter always do.[20] Thus, our the grammar for Lubukusu is given in (21). Again, the primary change compared to the baseline English grammar is that the constraints have been mirrored.

(21) Complementizer ϕ-agreement tier for Lubukusu

- Project: **D if [−EPP]**, all C
- Constraints:

$$\left\{\begin{array}{ll} X_{[+\phi]} \cdot X_{[+\phi]}, & X_{[-\phi]} \cdot X_{[-\phi]}, \\ \mathbf{X} \cdot \mathbf{X}_{[+\phi]}, & \mathbf{X}_{[-\phi]} \cdot \mathbf{X} \\ \rtimes \cdot \mathbf{X}_{[+\phi]}, & \mathbf{X}_{[-\phi]} \cdot \ltimes \end{array}\right\}$$

Note that Ermolaeva and Kobele (2022) arrive at a similar analysis, in which agreement is upward and targets the highest base-generated argument of the containing clause. However, their analysis requires successive overwriting of the morphology on the C head (the subject's value being the last to be written); as such, the present analysis could be considered simpler. An alternative TSL analysis would be to preserve the direction of probing, allowing D heads to search for and value agreeing C heads below them, as in Adger's (2003) treatment of affix hopping. Compared to the above analysis, we restore the relative positions of $+\phi$ and $-\phi$ in the TSL grammar and instead modify the featural content of the D and C heads in the lexicon. Ultimately, the direction of feature copying is the same under either analysis, suggesting that these two types of analysis are essentially notational variants.

4.3 *Multiple probes, one goal*

Until now, I have omitted treatment of several cases where multiple functional elements agree with the same DP. This included agreement on both the non-finite verb and finite auxiliary in Hindi, and agreement on both the verb and complementizer in West Flemish. Unlike in the example from Lubukusu, in these languages both agreeing elements are above the DP they agree with. In order to model such patterns with a single tier, we can adjust the tier constraints to allow

[20] Failure of agreement in hyperraising across an agreeing complementizer can be treated by selectively relaxing the constraints against lone probes while retaining those against lone goals, similar to our treatment of Icelandic.

multiple probes to appear in sequence, followed by a single shared goal. In essence, each probe obtains its value not from the source DP, but from the next closest agreeing functional head. This is analogous to the standard treatment of phonological harmony, in which feature spreading proceeds incrementally. We could also utilize separate tiers for each probe as we did for Lubukusu. However, we would lose the parallel with phonology; additionally, given that chain agreement is one of the basic predictions of a TSL-2 model, it is important to demonstrate that it can be handled with a single tier.

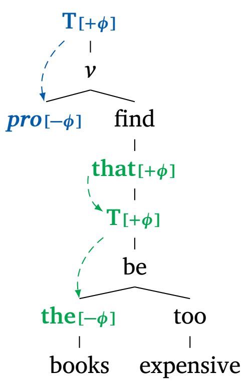


Figure 10:
Embedded C and T share a goal in West Flemish. In such cases, the tier grammar permits a sequence of probes followed by a goal

Keeping with the theme of complementizer treatment, let us return to the example from West Flemish in (18b), repeated in (22).[21] The structure assumed for this sentence is shown in Figure 10. While Flemish presumably has EPP movement, as well as V2 in main clauses, this is omitted for simplicity. As usual, the categories projected on the ϕ-agreement tier are D, T, and C. Upon extracting the spinal c-string for this example and projecting the tier, shown in (23), we obtain two adjacent chains of agreeing elements: a typical T/D pair in the main clause and a triple C/T/D in the embedded clause.

(22) Kvinden [$_{\text{CP}}$ dan die boeken [$_{\text{DegP}}$ te diere] zyn].
I.find that.PL the book.PL too expensive be.PL
'I find those books too expensive.'

[21] A very similar analysis can be used for concord phenomena of the sort mentioned in Section 4.2.

(23) C-string and tier projection for West Flemish chain agreement

- C-string: $\mathrm{T}_{[+\phi]} \cdot v \cdot \mathrm{D}_{[-\phi]} \cdot \mathrm{V} \cdot \mathrm{C}_{[+\phi]} \cdot \mathrm{T}_{[+\phi]} \cdot v \cdot \mathrm{D}_{[-\phi]} \cdot \textsc{deg} \cdot \mathrm{A}$
- ϕ tier: $\mathrm{T}_{[+\phi]} \cdot \mathrm{D}_{[-\phi]} \cdot \mathrm{C}_{[+\phi]} \cdot \mathrm{T}_{[+\phi]} \cdot \mathrm{D}_{[-\phi]}$

Our TSL grammars have always allowed multiple pairs of agreeing elements. What is new here is that a single agreement chain can contain more than two probes as long as they are ultimately followed by a goal. This is accomplished by removing the constraint $\mathrm{X}_{[+\phi]} \cdot \mathrm{X}_{[+\phi]}$.[22] Thus, the ϕ-agreement tier grammar for West Flemish is given in (24), with the removed constraint struck out.

(24) ϕ-agreement tier for West Flemish

- Project: all T, all D, all C
- Constraints: $\left\{ \begin{array}{ll} \cancel{\mathrm{X}_{[+\phi]} \cdot \mathrm{X}_{[+\phi]}}, & \mathrm{X}_{[-\phi]} \cdot \mathrm{X}_{[-\phi]}, \\ \mathrm{X}_{[+\phi]} \cdot \mathrm{X}, & \mathrm{X} \cdot \mathrm{X}_{[-\phi]}, \\ \mathrm{X}_{[+\phi]} \cdot \ltimes, & \rtimes \cdot \mathrm{X}_{[-\phi]} \end{array} \right\}$

As an alternative to the above analysis, we could also mark intermediate elements in the chain with both $+\phi$ and $-\phi$, explicitly signifying that they serve as both a probe and goal for agreement. In this case, the TSL grammar would be set up to allow substrings of the form $\mathrm{X}_{[+\phi]} \cdot \mathrm{X}_{[+\phi,-\phi]}$ and $\mathrm{X}_{[+\phi,-\phi]} \cdot \mathrm{X}_{[-\phi]}$, but not $\mathrm{X}_{[+\phi]} \cdot \mathrm{X}_{[+\phi]}$. The trouble with this approach is that some elements may need to be lexically specified as being available either as $\mathrm{X}_{[+\phi]}$ or $\mathrm{X}_{[+\phi,-\phi]}$ if they can occur both initially and chain-internally depending on the structural context. I am unaware of any cases in syntax where it is crucial to distinguish between probes which allow agreement to continue and those that do not, though it should be noted that the latter would correspond to so-called *icy targets* in phonology, which both harmonize and prevent harmony from spreading further. For now, it is simpler to treat all probes as equivalent.

At this point, the reader may be wondering about the opposite configuration, in which one probe agrees with several goals. We will allow exactly this in order to model a type of syntactic optionality, to be discussed in Section 4.4. Before that, I present an example which

[22] Note that the chains must be non-overlapping for this to work. In cases where chains of agreeing elements are interleaved with one another, multiple tiers are required.

summarizes several of the patterns that we have examined so far: invisibility, interaction with the other features, and shared goals. This is A′ agreement.

In previous examples, we projected DPs only if they were nominative, or only if they were EPP movers. These are both features normally associated with A-positions, but if we only project DPs bearing a certain A′ feature on the tier controlling ϕ-agreement, then we can derive agreement that targets A′ positions. A clear example comes from Dinka, a Nilotic language spoken in South Sudan. This language has a V2 clause structure in which agreement targets the initial DP, regardless of whether it is a subject, object, or oblique (Van Urk 2015, ch. 3). Examples with an initial subject (25a) and object (25b) are shown below. Additionally, agreement with Spec-CP occurs in relative clauses (of which *wh*-questions are one type) and intermediate movement sites; an example is given in (26).

(25) Dinka verb agreement with Spec-CP

a. Mòc à-cé yîin tîiŋ.
man 3SG-PRF.SV you see.NF
'The man has seen you.' (Van Urk 2015, ch. 4, 19b)

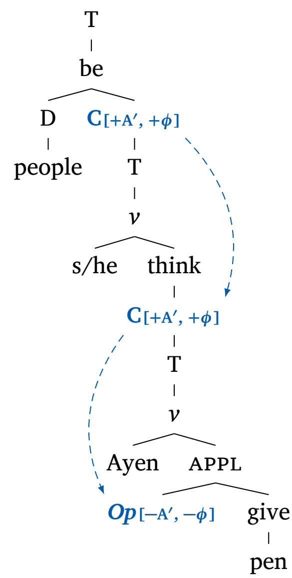
b. Yîin ∅-cíi môc tîiŋ.
you 2-PRF.OV man.GEN see.NF
'You, the man has seen.' (Van Urk 2015, ch. 4, 20a)

(26) Agreement in both matrix and embedded clause

Yè kɔ̂ɔc-kó [$_{CP}$ *Op* é-kè-yá ké tàak
be **people**.CS-which PST-**PL**-HAB.2SG 3PL think.NF
[$_{CP}$ è ___ é-kè-cíi Áyèn ké gâam gàlàm]]?
C PST-**PL**-PRF.OV Ayen.GEN 3PL give.NF pen
'Which people did (s)he think that Ayen had given a pen to?'
(Van Urk 2015, ch. 5, 14a)

Van Urk argues that there is a single generalized A′ feature that encompasses topicalization, relativization, and *wh*-movement. Additionally, there is some variation in whether embedded clauses allow, require, or disallow V2 (Van Urk 2015, p. 130). He proposes that the CP should be split into at least two levels, the lower of which, Fin, hosts V2. Following his lead, I will continue to refer to this head as C. It is also this head that is the locus of agreement. Accordingly, I assume that it bears the features $+\mathrm{A}'$ and $+\phi$. Note that I treat intermediate and final landing sites uniformly, mirroring our treatment of chain agreement, though this is not crucial to the analysis; an alternative would be to treat the complementizer morphology as the spell-out of a C head along a movement path, as proposed by Graf (2022b). A slightly simplified structure for example (26) which has been annotated accordingly is shown in Figure 11.[23]

Figure 11:
A′-agreement in Dinka. Each C head along the movement path agrees with the moving operator



The TSL analysis for this pattern is as follows. Since it is always the moving DP that agrees with C, we project only DPs bearing $-\mathrm{A}'$. C also appears on the tier since it agrees, but T neither agrees not blocks agreement, so it is omitted. The tier constraints are identical to our previous example of West Flemish. The full grammar is shown in (27), and example c-strings corresponding to sentences (25a) and (26) are shown in (28).

[23] The *ké* morpheme glossed as 3PL is omitted for simplicity. According to Van Urk, this morpheme occurs both as an independent pronoun and as a copy of movement at the edge of *v*P; in this case it is the latter.

(27) ϕ-agreement tier for Dinka

- Project: **D if [−A′]**, all C
- Constraints: $\left\{\begin{array}{ll} \sout{X_{[+\phi]} \cdot X_{[+\phi]}}, & X_{[-\phi]} \cdot X_{[-\phi]}, \\ X_{[+\phi]} \cdot X, & X \cdot X_{[-\phi]}, \\ X_{[+\phi]} \cdot \ltimes, & \rtimes \cdot X_{[-\phi]} \end{array}\right\}$

(28) C-string ϕ-agreement tiers for Dinka

a. Within-clause A′-movement
- C-string: $C_{[+A', +\phi]} \cdot T \cdot v \cdot D_{[-A', -\phi]} \cdot V \cdot D$
- Tier: $C_{[+A', +\phi]} \cdot D_{[-A', -\phi]}$

b. Long-distance A′-movement
- C-string: $T \cdot be \cdot D \cdot C_{[+A', +\phi]} \cdot T \cdot v \cdot D \cdot V \cdot C_{[+A', +\phi]} \cdot T \cdot v \cdot D \cdot \text{APPL} \cdot D_{[-A', -\phi]} \cdot V \cdot D$
- Tier: $C_{[+A', +\phi]} \cdot C_{[+A', +\phi]} \cdot D_{[-A', -\phi]}$

The present perspective cannot explain why ϕ-agreement occurs on C in Dinka, or why it is sensitive to A′-movement. But given that the distribution of these features is what it is, we correctly predict that their interaction results in an A′ locality profile for ϕ-agreement.

4.4 *Optionality and multiple goals*

In most of the preceding examples, there has been only a single agreement configuration available for each construction. But it is not rare to find instances of agreement which appear to be optional. As was briefly mentioned in Section 3.4, this happens even in English with agreement across existential *there*: for many speakers both singular and plural agreement are possible, especially when the logical subject is in an embedded clause, as in (6), repeated below as (29a).[24] In contrast, agreement with the logical subject is obligatory if *there* is not inserted (29b).

(29) Optional agreement across *there* in English existential clauses

a. There **seem(s)** to be <u>some squirrels</u> in the attic.

b. <u>Some squirrels</u> **seem(*s)** to be in the attic.

[24] This pattern should not be confused with reduced *'s*, as in *There's some squirrels in the attic*, which is acceptable even for speakers who do not accept %*There is some squirrels in the attic*.

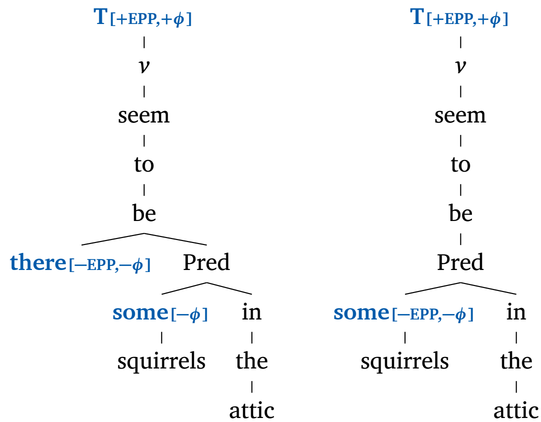


Figure 12: Interaction-satisfaction analysis of optional agreement in English existential clauses. Left: when *there* is inserted, T agrees both with *there* and the logical subject. Right: structure without *there*, where the logical subject is the only target for agreement

Such optionality presents a puzzle which has several plausible solutions, e.g. by positing multiple competing grammars (Kroch 1989). Even if we posit only a single grammar, there are several analytical options. Among these, the *interaction and satisfaction* theory of agreement (Deal 2015) fits well with the analyses presented so far. In the interaction-satisfaction theory, a probe can agree with multiple goals in a manner that is relativized to the individual probe. For each probe (EPP, ϕ, etc.), we specify its *interaction set*, which are the features that the probe agrees with, and its *satisfaction set*, which are the features that cause probing to stop. Upon spell-out, the probe may realize the features of any or all of the elements that it has agreed with in accordance with the morphology of the language. If there is more than one possible output, optionality results.

In the present case, we analyze optional agreement as resulting from agreement with both *there* and the logical subject, as illustrated in Figure 12. We further posit that *there* lacks some feature which is in the satisfaction set for the ϕ-probe on T: perhaps it has number features but lacks person features, for example. This allows T to agree with *there* but continue probing until it finds the logical subject. For present purposes, it does not matter exactly what is deficient about *there*, so for simplicity I will continue to treat all ϕ-features as a unit.

This brings us to the TSL analysis. This time, we project *there* on the tier just like any other DP, and we relax the tier constraints so that a ϕ-goal can be immediately followed by another ϕ-goal if the first goal is *there*, as shown in (30). As usual, c-strings for structures both with and without *there* are provided in (31). Only when *there* is selected is it possible for T to agree with two elements, so this is the only structure in which optional agreement occurs. Note that although both *there* and *some* are marked $-\phi$, this does not imply that they have the same value of ϕ; both are goals and therefore enter the derivation with separate values.

(30) ϕ-agreement tier for English (revised)

- Project: all T, all D, all C
- Constraints: as in (5), but allow there$_{[-\phi]}$ · D$_{[-\phi]}$

(31) C-strings and tier projections for optional agreement

a. With *there*

- C-string: T$_{[+\text{EPP},+\phi]}$ · *v* · seem · to · be · there$_{[-\text{EPP},-\phi]}$ · Pred · some$_{[-\phi]}$ · in · the · attic
- ϕ-agreement tier: T$_{[+\text{EPP},+\phi]}$ · there$_{[-\text{EPP},-\phi]}$ · some$_{[-\phi]}$

b. Without *there*

- C-string: T$_{[+\text{EPP},+\phi]}$ · *v* · seem · to · be · Pred · some$_{[-\text{EPP},-\phi]}$ · in · the · attic
- Tier: T$_{[+\text{EPP},+\phi]}$ · some$_{[-\text{EPP},-\phi]}$

In general, an analysis based on the interaction-satisfaction theory can be described as a TSL-2 pattern in which the probe is immediately followed by zero or more agreeing items with features in the interaction set but *not* the satisfaction set, possibly followed by one with features in the satisfaction set (regardless of whether it has any in the interaction set).[25] Thus, the class of agreement patterns which are TSL-2 potentially extends to many others which fall under the general schema of interaction and satisfaction, such as omnivorous agreement (Nevins 2011), also discussed by Deal (2015). For example, in the case of omnivorous number agreement where [PL] outranks [SG], the probe may be valued as [PL] if *any* DP in its search domain is [PL].

[25] Space prohibits me from providing a full analysis, but the basic idea is as follows. Let *P* denote a probe, *I* an interacting element, *S* a satisfying element, and *G* a normal goal, which both interacts with and satisfies the probe. We allow substrings such as {P · I, P · S, P · G, I · I, I · S, I · G} but not {S · I, S · G, G · I, G · G}. We must also distinguish *actual* interactors from *potential* interactors, perhaps with the same $-\phi$ diacritic used in this paper.

The TSL analysis of this pattern is essentially identical to the example of optional agreement across existential *there*. We place [SG] in the interaction set and [PL] in the satisfaction set. As before, $\mathrm{D}_{[-\phi]}$ may therefore be followed by another $\mathrm{D}_{[-\phi]}$ iff the first D is singular.

For completeness, I briefly mention an alternative approach to optional agreement, which is to allow certain items to project depending on whether or not they bear $-\phi$. For the present example, we would posit variants of *there* both with and without $-\phi$, and project only the variant bearing $-\phi$. Then, long-distance agreement would occur only when non-agreeing *there* is merged into the derivation. The disadvantage to this approach is that it violates the principle that potential agreeing elements should always project, which we have maintained in all preceding examples due to the pervasiveness of relativized minimality. However, formally there is nothing to prevent us from constructing a tier in this manner, and it may even be necessary for optional extraction morphology (Thomas Graf, p.c.).[26]

There is also a weakness to the interaction-satisfaction approach, which is that the tier constraints distinguish sets of lexical items in a more intricate manner than in previous examples. Unlike our treatment of multiple probes, the behavior of intermediate and final goals is different, and not controlled solely by the $-\phi$ feature. This potentially subverts the typology predicted by the present model, where the presence of $\pm\phi$ (or lack thereof) is the primary factor in the tier constraints. Again, it may the case that optionality should not even be handled within the syntactic grammar, but if we do so, there are several options which fit within the current framework; I leave a more thorough investigation of these and other options to future work.

4.5 *Summary*

To conclude this section, the agreement patterns analyzed in this paper are summarized again in Table 2, now including the specific tier projection functions for each case study. As before, the individual patterns are described in comparison to the baseline pattern of

[26] The idea of treating extraction morphology as constraints on a movement path goes back at least to Bouma *et al.* 2001.

Table 2: Summary of agreement patterns and their TSL-2 analysis

	Example	Tier projection	Tier constraints
a.	Subject-verb agr.	All T/D/C	Strict matching of $+\phi$ and $-\phi$
b.	Case-sensitive agr.	All T/C, D if [NOM]	(as in (a))
c.	Subject-orientation	All C, D if [−EPP]	(as in (a))
d.	A′ agreement	All C, D if [−A′]	(as in (a))
e.	Dative intervention	All T/C/D[NOM], D[DAT] if not [−EPP]	Non-agreeing dative may follow $+\phi$
f.	Upward agreement	(as in (a))	Swap order of $+\phi/-\phi$
g.	Chain agreement	(as in (a))	Allow sequential $+\phi$
h.	Multiple agreement	(as in (a))	Allow sequential $-\phi$

relativized minimality. For conciseness, only the Hindi variant of case-sensitive agreement is included; additionally, the two components of the Dinka complementizer agreement pattern from Section 4.3 (A′ agreement and multiple probes) have been factored out and listed separately.

Having seen how the proposed model works in a variety of languages, we can now better assess the match between its formal capabilities and the observed typology. Recall from Section 2.1 that by restricting ourselves to TSL, many conceptually simple yet linguistically unnatural constraints become impossible to implement, at least in full generality. Sometimes, it is possible to construct limited counterexamples. For example, threshold counting can be simulated by choosing a tier which contains just the elements of interest and a window size large enough to contain the maximum number we wish to count to. This could be used to construct a language in which a verb is plural iff any of the first *four* DPs in its c-command domain is plural. The restriction to TSL-2 helps to further rule out such tricks.

Indeed, as argued by McMullin and Hansson (2016) and Graf (2022b), TSL-2 gives us exactly the kind of locality restrictions characteristic of natural language: the presence of even a single blocker breaks any long-distance dependency. Some dependencies are strictly local, while others lack blockers altogether, but what we do not find are patterns in which at most one blocker, or two, or three, may be tolerated but no more.[27] Other obvious manipulations of a TSL-2 grammar, such as mirroring the constraints, and allowing adjacent pairs of like elements, likewise correspond to real agreement phenomena.

Even so, as several reviewers of this paper remarked, the freedom of the tier projection function to include or exclude any symbol according to its label seems to overgenerate. For example, we could define a function that projects a random assortment of D heads, rather than all of them. This seems unavoidable since the computational system has no knowledge of the substantive interpretation of the element labels. Furthermore, the existence of lexical exceptions alongside productive generalizations suggests that only the acquisition theory can correctly restrict the set of possible tier projections.[28] It is for these reasons, among others, that I have stressed that the TSL-2 hypothesis is only one component of a complete theory.

5 CONCLUSION

We have seen that a wide variety of agreement phenomena are in fact variations on a simple theme: a TSL-2 pattern which involves the pairing of probes and goals for agreement. This simple model predicts the prevalence of relativized minimality as well as variation in the sets of invisible and blocking elements. Variation across languages can be accounted for using slight adjustments in the tier projection and the constraints in a way that closely tracks the logical possibilities afforded by the formalism.

It is worth reiterating that we do not expect every possible formal pattern to be attested due to the limited number of existing languages

[27] The literature contains theories in which at most one blocker may be crossed, including subjacency (Chomsky 1973) and some versions of phase theory (Chomsky 2001, 2008). Such theories are enmeshed in many auxiliary assumptions, such as successive cyclic movement, making them difficult to evaluate.

[28] The Tolerance Principle (Yang 2016) seems to me to be eminently compatible with the subregular perspective. There already exists some work which attempts to integrate this idea with the learning of TSL-2 grammars. See Belth 2023 for an example from phonology and Hanson 2024b regarding syntax.

and the other factors influencing typology. As such, it is informative that so many predicted patterns represent actual agreement phenomena. Also, as I have pointed out several times, the formal parallel between syntactic agreement and phonological harmony is particularly close. This can be explained as follows: both are feature-copying phenomena, and both involve the same TSL-2 computations, so both admit the same basic range of formal variation.

Since I have only treated a handful of illustrative examples, additional work is needed to strengthen the claim that individual agreement patterns are in fact TSL-2 over c-strings. In particular, agreement patterns with constraints on multiple tiers merit a detailed examination, in order to confirm that the full grammar is MTSL. It would also be prudent to formalize larger fragments of the grammars of individual languages. And yet, as we have seen several times, the question of whether a given agreement pattern is TSL-2 tends to be robust to differences in the precise details of the analysis. Because of this, it is mainly the empirical facts that need to be scrutinized carefully.

Several additional questions remain open. As mentioned previously, some instances of feeding/bleeding of agreement by movement may require knowledge of the exact position of movers at different points in the derivation. At the same time, it would appear that not all instances of movement feed agreement (similar to how some moved elements undergo semantic reconstruction), as we saw with hyperraising and complementizer agreement in Lubukusu. This suggests the need for a model which tracks both the base and subsequent positions of movers. Coordination also introduces difficulties such as first/last conjunct agreement, which appear at first glance to be beyond the scope of the c-string model. A more complete model may require the ability to look a short distance into complex left branches, as discussed by Graf and De Santo (2019). Alternatively, we might use feature percolation to bring the correct information up to the top of the structure so that it becomes visible to the containing c-string.

Finally, it is unclear what the exact relation is between agreement in the strict sense and similar long-distance dependencies such as NPI-licensing. The Minimalist literature contains many claims of the form “phenomenon X should be reduced to operation Y”, where Y is typically Merge/Move/Agree. I have suggested that this might be the wrong level of granularity, and instead, we should consider movement, agreement, case, and so on to each be instances of TSL computations, and likewise for other conceptually distinct phenomena. Now that we have evidence that all of these patterns are related by their computational complexity, it should be possible to factor out this property in order to tell what, if any, differences remain.

ACKNOWLEDGMENTS

This work was partly supported by NSF Grant BCS-1845344 and by the Institute for Advanced Computational Science at Stony Brook University. Many thanks to Thomas Graf and three anonymous reviewers for their extremely detailed feedback on several drafts of this paper, which resulted in a much improved presentation. I also thank Sandhya Sundaresan, Tom McFadden, John Bailyn, Gary Thoms, and audiences at the NYU Linguistics Brown Bag, CLS 60, and the Workshop on Myopia in Grammar at Leipzig University.

REFERENCES

David ADGER (2003), *Core syntax: A Minimalist approach*, Oxford University Press, doi:10.1093/oso/9780199243709.001.0001.

Avery ANDREWS (1982), The representation of case in Modern Icelandic, in Joan BRESNAN, editor, *The mental representation of grammatical relations*, pp. 427–503, MIT Press.

Joseph AOUN and Dominique SPORTICHE (1982), On the formal theory of government, *The Linguistic Review*, 2(3):211–236, doi:10.1515/tlir-1983-020303.

Mark BAKER (2008), *The syntax of agreement and concord*, Cambridge University Press, doi:10.1017/CBO9780511619830.

Mark BAKER and Nadya VINOKUROVA (2010), Two modalities of case assignment: Case in Sakha, *Natural Language & Linguistic Theory*, 28(3):593–642, doi:10.1007/s11049-010-9105-1.

Susana BÉJAR and Milan REZAC (2009), Cyclic agree, *Linguistic Inquiry*, 40(1):35–73, doi:10.1162/ling.2009.40.1.35.

Caleb BELTH (2023), Towards a learning-based account of underlying forms: A case study in Turkish, in *Proceedings of the Society for Computation in Linguistics 2023*, pp. 332–342, University of Massachusetts Amherst, doi:10.7275/BC8Q-VJ22.

Jonathan BOBALJIK (2008), Where's phi? Agreement as a postsyntactic operation, in Daniel HARBOUR, David ADGER, and Susana BÉJAR, editors, *Phi Theory*, pp. 295–328, Oxford University Press, doi:10.1093/oso/9780199213764.003.0010.

Marisa Ferrara BOSTON, John T. HALE, and Marco KUHLMANN (2010), Dependency structures derived from Minimalist Grammars, in *Proceedings of the 10th and 11th Biennial Conference on The Mathematics of Language*, pp. 1–12, Springer-Verlag, Berlin, Heidelberg, doi:10.1007/978-3-642-14322-9_1.

Gosse BOUMA, Robert MALOUF, and Ivan A. SAG (2001), Satisfying constraints on extraction and adjunction, *Natural Language & Linguistic Theory*, 19(1):1–65, doi:10.1023/a:1006473306778.

Joan BRESNAN, Ronald M. KAPLAN, Stanley PETERS, and Annie ZAENEN (1982), Cross-serial dependencies in Dutch, *Linguistic Inquiry*, 13(4):613–635, `https://www.jstor.org/stable/4178298`.

Phillip Alexander BURNESS, Kevin James MCMULLIN, Jane CHANDLEE, Phillip BURNESS, and Kevin MCMULLIN (2021), Long-distance phonological processes as tier-based strictly local functions, *Glossa: A Journal of General Linguistics*, 6(1), doi:10.16995/glossa.5780.

Noam CHOMSKY (1957), *Syntactic structures*, De Gruyter Mouton, doi:10.1515/9783112316009.

Noam CHOMSKY (1959), On certain formal properties of grammars, *Information and Control*, 2(2):137–167, doi:10.1016/S0019-9958(59)90362-6.

Noam CHOMSKY (1973), Conditions on transformations, in Stephen R. ANDERSON and Paul KIPARSKY, editors, *A Festschrift for Morris Halle*, pp. 232–286, Holt, Rinehart and Winston.

Noam CHOMSKY (1995), *The Minimalist Program*, MIT Press, doi:10.7551/mitpress/9780262527347.001.0001.

Noam CHOMSKY (2000), Minimalist inquiries: The framework, in Roger MARTIN, David MICHAELS, and Juan URIAGEREKA, editors, *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, pp. 89–155, MIT Press.

Noam CHOMSKY (2001), Derivation by phase, in Michael KENSTOWICZ, editor, *Ken Hale: A Life in Language*, pp. 1–52, MIT Press, doi:10.7551/mitpress/4056.003.0004.

Noam CHOMSKY (2008), On Phases, in *Foundational Issues in Linguistic Theory: Essays in Honor of Jean-Roger Vergnaud*, pp. 133–166, MIT Press, doi:10.7551/mitpress/7713.003.0009.

Jessica COON and Clint PARKER (2019), Case interactions in syntax, in *Oxford Research Encyclopedia of Linguistics*, Oxford University Press, doi:10.1093/acrefore/9780199384655.013.315.

Aniello De Santo and Thomas Graf (2019), Structure sensitive tier projection: Applications and formal properties, in Raffaella Bernardi, Gregory Kobele, and Sylvain Pogodalla, editors, *Formal Grammar: 24th International Conference*, pp. 35–50, Springer, doi:10.1007/978-3-662-59648-7_3.

Amy Rose Deal (2009), The origin and content of expletives: Evidence from "selection", *Syntax*, 12(4):285–323, doi:10.1111/j.1467-9612.2009.00127.x.

Amy Rose Deal (2015), Interaction and satisfaction in ϕ-agreement, in *Proceedings of NELS 45*, pp. 179–192, GLSA Publications.

Michael Diercks (2013), Indirect agree in Lubukusu complementizer agreement, *Natural Language & Linguistic Theory*, 31(2):357–407, doi:10.1007/s11049-013-9187-7.

Marina Ermolaeva (2018), Morphological agreement in Minimalist Grammars, in Annie Foret, Reinhard Muskens, and Sylvain Pogodalla, editors, *Formal Grammar 2018: 23rd International Conference*, pp. 20–36, Springer, doi:10.1007/978-3-662-56343-4_2.

Marina Ermolaeva and Gregory M. Kobele (2023), Formal properties of agreeing Minimalist Grammars, in *Proceedings of the Society for Computation in Linguistics 2023*, doi:10.7275/KQY6-0737.

Marina Ermolaeva and Gregory M. Kobele (2022), Agree as information transmission over dependencies, *Syntax*, 25(4):466–507, doi:10.1111/synt.12240.

John Goldsmith (1976), *Autosegmental phonology*, Ph.D. thesis, Massachusetts Institute of Technology, `https://dspace.mit.edu/handle/1721.1/16388`.

Thomas Graf (2013), *Local and transderivational constraints in syntax and semantics*, Ph.D. thesis, University of California, Los Angeles, `https://thomasgraf.net/output/graf13thesis.html`.

Thomas Graf (2018), Why movement comes for free once you have adjunction, in *Proceedings of CLS 53*, pp. 117–136, The Chicago Linguistic Society.

Thomas Graf (2020), Monotonicity in syntax, in Dun Deng, Fenrong Liu, Mingming Liu, and Dag Westerståhl, editors, *Monotonicity in Logic and Language*, pp. 35–53, Springer, doi:10.1007/978-3-662-62843-0_3.

Thomas Graf (2022a), Subregular linguistics: bridging theoretical linguistics and formal grammar, *Theoretical Linguistics*, 48(3–4):145–184, doi:10.1515/tl-2022-2037.

Thomas Graf (2022b), Typological implications of tier-based strictly local movement, in *Proceedings of the Society for Computation in Linguistics 2022*, pp. 184–193, University of Massachusetts Amherst Libraries, doi:10.7275/gb65-ht31.

Thomas GRAF (2023), Subregular tree transductions, movement, copies, traces, and the ban on improper movement, in *Proceedings of the Society for Computation in Linguistics 2023*, pp. 289–299, University of Massachusetts Amherst Libraries, doi:10.7275/TK1N-Q855.

Thomas GRAF and Aniello DE SANTO (2019), Sensing tree automata as a model of syntactic dependencies, in *Proceedings of the 16th Meeting on the Mathematics of Language*, pp. 12–26, University of Massachusetts Amherst Libraries, doi:10.18653/v1/W19-5702.

Thomas GRAF and Kalina KOSTYSZYN (2021), Multiple wh-movement is not special: The subregular complexity of persistent features in Minimalist Grammars, in *Proceedings of the Society for Computation in Linguistics 2021*, pp. 275–285, University of Massachusetts Amherst Libraries, doi:10.7275/2dfv-v219.

Thomas GRAF and Connor MAYER (2018), Sanskrit n-retroflexion is input-output tier-based strictly local, in *Proceedings of SIGMORPHON 2018*, pp. 151–160, Association for Computational Linguistics, doi:10.18653/v1/W18-5817.

Thomas GRAF and Nazila SHAFIEI (2019), C-command dependencies as TSL string constraints, in *Proceedings of the Society for Computation in Linguistics 2019*, pp. 205–215, University of Massachusetts Amherst Libraries, doi:10.7275/4rrx-x488.

Kenneth HANSON (2023), A TSL analysis of Japanese case, in *Proceedings of the Society for Computation in Linguistics 2023*, pp. 15–24, University of Massachusetts Amherst Libraries, doi:10.7275/xqhr-r404.

Kenneth HANSON (2024a), Strict locality in syntax, in *Proceedings of CLS 59*, pp. 131–145, The Chicago Linguistic Society.

Kenneth HANSON (2024b), Tiers, paths, and syntactic locality: The view from learning, in *Proceedings of the Society for Computation in Linguistics 2024*, pp. 107–116, University of Massachusetts Amherst Libraries, doi:10.7275/scil.2135.

Jeffrey HEINZ (2018), The computational nature of phonological generalizations, in Larry M. HYMAN and Frans PLANK, editors, *Phonological Typology*, number 23 in Phonetics and Phonology, pp. 126–195, De Gruyter Mouton, doi:10.1515/9783110451931-005.

Jeffrey HEINZ, Chetan RAWAL, and Herbert G. TANNER (2011), Tier-based strictly local constraints for phonology, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human language technologies*, pp. 58–64.

Anders HOLMBERG and Thorbjörg HRÓARSDÓTTIR (2003), Agreement and movement in Icelandic raising constructions, *Lingua*, 113(10):997–1019, doi:10.1016/S0024-3841(02)00162-6.

Riny HUYBREGTS (1976), Overlapping dependencies in Dutch, *Utrecht Working Papers in Linguistics*, 1:24–65.

Riny HUYBREGTS (1984), The weak inadequacy of context-free phrase structure grammars, in Ger J. DE HAAN, Mieke TROMMELEN, and Wim ZONNEVELD, editors, *Van Periferie naar Kern*, pp. 81–99, Foris.

Adam JARDINE and Kevin MCMULLIN (2017), Efficient learning of tier-based strictly k-local languages, in Frank DREWES, Carlos MARTÍN-VIDE, and Bianca TRUTHE, editors, *Language and Automata Theory and Applications*, pp. 64–76, Springer, Cham, doi:10.1007/978-3-319-53733-7_4.

Aravind K. JOSHI (1985), Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions?, in David R. DOWTY, Lauri KARTTUNEN, and Arnold M. ZWICKY, editors, *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, pp. 206–250, Cambridge University Press, doi:10.1017/CBO9780511597855.007.

Ronald M. KAPLAN and Martin KAY (1994), Regular models of phonological rule systems, *Computational Linguistics*, 20(3):331–378, `http://www.aclweb.org/anthology/J94-3001.pdf`.

Stefan KEINE (2019), Selective opacity, *Linguistic Inquiry*, 50(1):13–62, doi:10.1162/ling_a_00299.

Gregory M. KOBELE (2002), Formalizing mirror theory, *Grammars*, 5(3):177–221, doi:10.1023/A:1022104104992.

Gregory M. KOBELE (2012), Eliding the derivation: A minimalist formalization of ellipsis, in *Proceedings of the International Conference on Head-Driven Phrase Structure Grammar*, pp. 409–426, CSLI Publications, doi:10.21248/hpsg.2012.23.

Anthony S. KROCH (1989), Reflexes of grammar in patterns of language change, *Language Variation and Change*, 1(3):199–244, doi:10.1017/S0954394500000168.

Ivona KUČEROVÁ (2016), Long-distance agreement in Icelandic: locality restored, *The Journal of Comparative Germanic Linguistics*, 19(1):49–74, doi:10.1007/s10828-016-9077-6.

Dakotah LAMBERT (2021), Grammar interpretations and learning TSL online, in Jane CHANDLEE, Rémi EYRAUD, Jeff HEINZ, Adam JARDINE, and Menno VAN ZAANEN, editors, *Proceedings of the Fifteenth International Conference on Grammatical Inference*, volume 153 of *Proceedings of Machine Learning Research*, pp. 81–91, PMLR, `https://proceedings.mlr.press/v153/lambert21a.html`.

Dakotah LAMBERT (2023), Relativized adjacency, *Journal of Logic, Language and Information*, 32(4):707–731, doi:10.1007/s10849-023-09398-x.

Dakotah LAMBERT, Jonathan RAWSKI, and Jeffrey HEINZ (2021), Typology emerges from simplicity in representations and learning, *Journal of Language Modelling*, 9(1):151–194, doi:10.15398/jlm.v9i1.262.

Dakotah LAMBERT and James ROGERS (2020), Tier-based strictly local stringsets: Perspectives from model and automata theory, in *Proceedings of the Society for Computation in Linguistics 2020*, pp. 159–166, University of Massachusetts Amherst, Amherst, MA, `https://aclanthology.org/2020.scil-1.21`.

Anoop Kumar MAHAJAN (1990), *The A/A-bar distinction and movement theory*, Ph.D. thesis, Massachusetts Institute of Technology, `https://dspace.mit.edu/handle/1721.1/13650`.

Connor MAYER (2021), Capturing gradience in long-distance phonology using probabilistic tier-based strictly local grammars, in *Proceedings of the Society for Computation in Linguistics 2021*, pp. 39–50, University of Massachusetts Amherst Libraries, doi:10.7275/231q-p480.

Connor MAYER and Travis MAJOR (2018), A challenge for tier-based strict locality from Uyghur backness harmony, in *Formal Grammar 2018: 23rd International Conference*, pp. 62–83, Springer, doi:10.1007/978-3-662-57784-4_4.

Kevin MCMULLIN (2016), *Tier-based locality in long-distance phonotactics: learnability and typology*, Ph.D. thesis, University of British Columbia.

Kevin MCMULLIN and Gunnar Ólafur HANSSON (2016), Long-distance phonotactics as tier-based strictly 2-local languages, in *Proceedings of the 2014 Annual Meeting on Phonology*, Linguistic Society of America, doi:10.3765/amp.v2i0.3750.

Robert MCNAUGHTON and Seymour A. PAPERT (1971), *Counter-free automata*, MIT Press.

Andrew NEVINS (2010), *Locality in vowel harmony*, MIT Press, doi:10.7551/mitpress/9780262140973.001.0001.

Andrew NEVINS (2011), Multiple agree with clitics: Person complementarity vs. omnivorous number, *Natural Language & Linguistic Theory*, 29:939–971, doi:10.1007/s11049-011-9150-4.

Elissa L. NEWPORT and Richard N. ASLIN (2004), Learning at a distance I. Statistical learning of non-adjacent dependencies, *Cognitive Psychology*, 48(2):127–162, doi:10.1016/S0010-0285(03)00128-2.

David PESETSKY and Esther TORREGO (2007), The syntax of valuation and the interpretability of features, in Simin KARIMI, Vida SAMIIAN, and Wendy K. WILKINS, editors, *Phrasal and Clausal Architecture: Syntactic Derivation and Interpretation*, Benjamins, doi:10.1075/la.101.14pes.

Maria POLINSKY and Eric POTSDAM (2001), Long-distance agreement and topic in Tsez, *Natural Language & Linguistic Theory*, 19(3):583–646, doi:10.1023/A:1010757806504.

Omer PREMINGER (2014), *Agreement and its failures*, MIT Press, doi:10.7551/mitpress/9780262027403.001.0001.

Tanya REINHART (1981), Definite NP anaphora and c-command domains, *Linguistic Inquiry*, 12(4):605–635, https://www.jstor.org/stable/4178245.

Luigi RIZZI (1990), *Relativized minimality*, MIT Press.

Brian ROARK and Richard SPROAT (2007), *Computational approaches to morphology and syntax*, Oxford University Press, doi:10.1093/oso/9780199274772.001.0001.

James ROGERS (1997), Strict LT_2 : Regular :: Local : Recognizable, in Christian RETORÉ, editor, *Logical Aspects of Computational Linguistics: First International Conference, LACL '96 (Selected Papers)*, volume 1328 of *Lectures Notes in Computer Science/Lectures Notes in Artificial Intelligence*, pp. 366–385, Springer, doi:10.1007/BFb0052167.

James ROGERS, Jeffrey HEINZ, Gil BAILEY, Matt EDLEFSEN, Molly VISSCHER, David WELLCOME, and Sean WIBEL (2010), On languages piecewise testable in the strict sense, in Christian EBERT, Gerhard JÄGER, and Jens MICHAELIS, editors, *The Mathematics of Language*, volume 6149 of *Lecture Notes in Artifical Intelligence*, pp. 255–265, Springer, doi:10.1007/978-3-642-14322-9_19.

James ROGERS, Jeffrey HEINZ, Margaret FERO, Jeremy HURST, Dakotah LAMBERT, and Sean WIBEL (2013), Cognitive and sub-regular complexity, in Glyn MORRILL and Mark-Jan NEDERHOF, editors, *Formal Grammar: 17th and 18th International Conferences*, volume 8036 of *Lecture Notes in Computer Science*, pp. 90–108, Springer, doi:10.1007/978-3-642-39998-5_6.

Marcel Paul SCHÜTZENBERGER (1965), On finite monoids having only trivial subgroups, *Information and Control*, 8(2):190–194, doi:10.1016/S0019-9958(65)90108-7.

Stuart M. SHIEBER (1985), Evidence against the context-freeness of natural language, in Walter J. SAVITCH, Emmon BACH, William MARSH, and Gila SAFRAN-NAVEH, editors, *The formal complexity of natural language*, pp. 320–334, Springer, doi:10.1007/BF00630917.

Halldór Ármann SIGURÐSSON and Anders HOLMBERG (2008), Icelandic dative intervention: Person and number are separate probes, in Roberta D'ALESSANDRO, Susann FISCHER, and Gunna Hrafn HRAFNBJARGARSON, editors, *Agreement restrictions*, pp. 176–194, De Gruyter Mouton, doi:10.1515/9783110207835.251.

Imre SIMON (1975), Piecewise testable events, in *Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages*, pp. 214–222, Springer-Verlag, doi:10.1007/3-540-07407-4_23.

Edward P. STABLER (1997), Derivational Minimalism, in Christian RETORE, editor, *Logical Aspects of Computational Linguistics*, Springer, doi:10.1007/BFb0052152.

Edward P. STABLER (2011), Computational perspectives on Minimalism, in Cedric BOECKX, editor, *Oxford Handbook of Linguistic Minimalism*, pp. 617–643, Oxford University Press, doi:10.1093/oxfordhb/9780199549368.013.0027.

Hoskuldur THRAINSSON (2007), *The syntax of Icelandic*, Cambridge University Press, doi:10.1017/CBO9780511619441.

Coppe VAN URK (2015), *A uniform syntax for phrasal movement: A Dinka Bor case study*, Ph.D. thesis, Massachusetts Institute of Technology, https://dspace.mit.edu/handle/1721.1/101595.

Mai Ha VU, Nazila SHAFIEI, and Thomas GRAF (2019), Case assignment in TSL syntax: A case study, in *Proceedings of the Society for Computation in Linguistics 2019*, pp. 267–276, University of Massachusetts Amherst Libraries, doi:10.7275/sywz-xw23.

Charles YANG (2016), *The price of linguistic productivity: How children learn to break the rules of language*, MIT Press.

Hedde ZEIJLSTRA (2012), There is only one way to agree, *The Linguistic Review*, 29(3):491–539, doi:10.1515/tlr-2012-0017.

Kenneth Hanson
0009-0005-4243-3976
mail@kennethhanson.net

Stony Brook University

Kenneth Hanson (2025), *Tier-based strict locality and the typology of agreement*, Journal of Language Modelling, 13(1):43–97
https://dx.doi.org/10.15398/jlm.v13i1.411

Against successive cyclicity: A proof-theoretic account of extraction pathway marking

Yusuke Kubota[1] *and Robert Levine*[2]
[1] NINJAL
[2] Ohio State University

ABSTRACT

This paper proposes a novel analysis of extraction pathway marking in Type-Logical Grammar, taking advantage of proof-theoretic properties of logical proofs whose empirical application has so far been underexplored. The key idea is to allow certain linguistic expressions to be sensitive to the intermediate status of a syntactic proof. The relevant conditions can be stated concisely as constraints at the level of the proof term language, formally a special type of λ-calculus. The proposed analysis does not have any direct analog to either of the two familiar techniques for analyzing extraction pathway marking, namely, successive cyclic movement in derivational syntax and the SLASH feature percolation in HPSG.

Moreover, the 'meaning-centered' perspective that naturally emerges from this new analysis is conceptually revealing: on this approach, extraction pathway marking essentially boils down to a strategy that certain languages employ to overtly flag the existence of a semantic variable inside a partially derived linguistic expression whose interpretation is dependent on a higher-order operator that is located in a larger structure.

Keywords: *long-distance dependencies, extraction pathway marking, successive cyclicity, phase, locality, Type-Logical Grammar, proof theory*

1 INTRODUCTION

A widely entertained assumption in generative syntax holds that the long-distance movement operation is 'successively cyclic' (Chomsky 1973, 1977). This assumption is a fundamental part of the theory in virtually all avatars of derivational syntax since the 1970s, and is standardly taken to constitute an explanation for why movement operations in natural language are constrained in the way they appear to be, reflected in phenomena such as island constraints (see Section 2 for more on this).[1] The status of islands has been questioned much in the recent literature, but successive cyclicity is taken to receive more direct empirical evidence from typologically diverse languages in the so-called *extraction pathway marking* (EPM) phenomena (Kayne and Pollock 1978; McCloskey 1979; Chung 1982; Zaenen 1983; Borsley 2010; van Urk and Richards 2015, among others). In EPM, a syntactically displaced expression (such as the fronted *wh*-phrase in *wh*-questions)[2] induces overtly visible effects at the intermediate landing sites of a chain of movement linking the filler and the gap.

This can be illustrated most clearly by the choice of complementizer in Irish reported in McCloskey 1979. For expository convenience, we illustrate the pattern by a pseudo-language called Iringlish, which is like Irish in having the relevant distinction of two complementizers but is identical to English in all other respects.[3] As shown in (1), Iringlish (or Irish) has two complementizers *aL* and *goN* that are in complementary distribution: *aL* is used when the complementizer position

[1] See Pullum (1992) for an insightful and critical survey of the theoretical status of 'transformational cycle' in the history of generative grammar.

[2] Note that the 'filler' is not always overt, as in the case of zero relatives in English *the book I thought John read __*.

[3] The complementary distribution of the *a*- and *g*- series of Irish complementizers has been extensively discussed in James McCloskey's work (see, e.g., McCloskey 1979, 1990, 2002); for an alternative view of the morphosyntactic status of these markers, see Sells (1984). We follow McCloskey's notation in his use of upper-case letters to identify the lenition- and nasalization-triggering effects of these markers as part of the Irish Gaelic mutation system; for a recent overview of this pattern across the Celtic languages, see Iosad 2023.

is crossed by *wh*-movement (here, the covert movement of the relativization operator *Op*); *goN* appears elsewhere.

(1) a. the man *Op* **aL** [I said __ **aL** [I thought __ **aL** [__ would be there]]]

b. the man *Op* **aL** [he said __ **aL** [__ thought **goN** [he would be there]]]

The goal of the present paper is to propose an alternative account of extraction pathway marking in a proof-theoretic variant of categorial grammar (CG) known as Type-Logical Grammar (TLG). Detailed analyses of EPM effects are currently lacking in the CG literature.[4] The analysis we argue for is novel in that it does not recognize either successive cyclic movement or feature percolation of the sort utilized in the non-movement analyses of extraction pathway marking (Bouma *et al.* 2001). This surprising result comes from trying to analyze this phenomenon in a theory in which neither device is native to the underlying architecture.

The new analysis we advocate capitalizes on the proof-theoretic perspective inherent to TLG, but its core idea is arguably more general and has clear connections to the leading ideas behind many proposals within mainstream syntax (at least at an abstract level). The key claim of the present paper is that extraction pathway marking can be best understood as a 'strategy' that the grammar of some languages employs in making the intermediate (or 'incomplete') status of linguistic composition (formalized as proofs in TLG) visible in surface syntax. Making direct reference to the structure of proofs is a controversial move within the linguistic tradition of TLG (or categorial grammar research more generally). We argue that this is precisely what is needed to account for extraction pathway marking, and that by making this move, we gain conceptual clarity: the proof-theoretic perspective *predicts* the existence of extraction pathway marking in natural language, in the sense that the phenomenon exploits exactly what the grammar offers as available resource, in a conceptually simple way.

[4] The only exception we are aware of is Kubota and Levine (2020), which – as the authors themselves admit – is essentially a clumsy rendering of the HPSG-style feature percolation analysis by Bouma *et al.* (2001) within TLG.

We believe that this somewhat contentious claim would be of interest to many syntacticians and semanticists, in both 'mainstream' and 'non-mainstream' approaches. To cater to different types of audience with different backgrounds, the presentation of the material in what follows is somewhat nonstandard: after reviewing the history of the notion of cyclicity in mainstream syntax in Section 2, we present the key component of the analysis in informal terms in Section 3. This is followed by a self-contained quick review of TLG in Section 4. Section 5 then presents the analysis in full detail (Section 5.1), and puts it into perspective in relation to three larger issues: Section 5.2 examines a wider range of languages and addresses a recent claim by van Urk and Richards (2015) and van Urk (2020), according to which *both* the movement-type mechanism and the feature percolation-type mechanism are needed for a proper analysis of EPM; Section 5.3 briefly discusses implications for other phenomena pertaining to cyclicity such as reconstruction effects; Section 5.4 offers a brief comparison with a feature percolation analysis in HPSG. Section 6 concludes the paper.

2 THE STATUS OF THE NOTION OF CYCLICITY IN DERIVATIONAL SYNTAX

In this section, we review the theoretical background on the notion of cyclicity (Section 2.1) and the empirical literature on extraction pathway marking (Section 2.2). The empirical and theoretical literature is entangled in a quite complex manner, as this topic directly pertains to one of the core issues in modern syntax: the proper characterization of long-distance dependencies in natural language. The main points we aim to establish in this section are the following:

(i) The notion of cyclicity is standardly taken to constitute a fundamental principle from which various 'locality' conditions (such as island sensitivity) are supposed to follow, but this syntax-oriented perspective has come under increasing scrutiny over the years.

(ii) Many of the reported cases of alleged 'evidence' for EPM/cyclicity are also controversial since they are often based on incorrect empirical generalizations or lack proper comparison with alternative analyses that don't rely on cyclicity.

It should be noted at the outset that by making these critical remarks on the previous syntactic literature, we do not mean to claim that there is nothing that needs to be encoded in syntax to account for the EPM patterns. Rather, our point is merely that the notion of cyclicity merits reconceptualization, and that empirical evidence for it should be scrutinized at the same time in such critical rethinking. We argue that the semantically-oriented reconceptualization we propose in Section 3 (and demonstrate further in Section 5) does offer a new perspective on the relevant empirical facts themselves, by identifying this phenomenon as an overt manifestation of the intermediate status of linguistic composition of 'variable-containing' expressions.

A brief history 2.1

The notion of cyclicity as the basis for long-distance dependencies has its origins in Chomsky's (1973) proposal to derive Ross's (1967) Complex Noun Phrase Constraint (CNPC) from more general principles. Chomsky specified certain syntactic positions, specifically S and NP, as *bounding nodes* and stipulated that no more than one of such bounding nodes could be crossed at a time. Further extensions of this perspective in Chomsky (1981, 1986) led to the so-called 'Barriers' model, in which the configurational restrictions on movement were made to follow from the distinction between constituents which are 'lexically selected' and those which are not. But irrespective of precisely how the configurational restrictions on extraction were defined, the fundamental basis for such restrictions has always been entangled with the key premise in Chomsky (1973) that long-distance dependencies are an epiphenomenon of local movements chained together through unbounded iterations, and that restrictions on such dependencies are due to syntactic conditions which break such cyclically created chains.

From the early days on, it has been recognized that the mere compatibility of the distribution of islands with one or another set of syntactic configurations does not on its own amount to positive evidence for some particular set of principles of the sort Chomsky proposed. For this reason, the discovery of morphosyntactic or phonological effects that mirror the pattern of cyclic movement via bounding nodes was important. Such 'syntactic reflexes' of cyclicity have been called extraction pathway marking (EPM) effects. See Clements *et al.* 1983 and Zaenen 1983 for earliest theoretical discussions. Reported cases of EPM in the early literature include complementizer choice in Irish (McCloskey 1979), subject-auxiliary inversion in French (Kayne and Pollock 1978) and verb agreement in Chamorro (Chung 1982).

Although the underlying architecture of the derivational theory has changed significantly over the years, especially after the advent of the Minimalist Program (MP), the idea behind cyclic movement has essentially survived to date. In the MP formulation, the notion of 'phase' – a syntactic domain where the complement of the functional head is transferred to PF at certain points in the derivation – has technically replaced the older variants of the idea of cyclic movement through certain syntactically designed positions.

Just as the main motivation of Chomsky's (1973) original proposal was to reduce some of the island effects to more general notions, the main theoretical import of the notion of phase is understood to lie in the fact that it serves as the underlying principle from which superficially observable phenomena such as island effects are to be derived. And just as in the Transformational era, the EPM effects continue to be regarded as major empirical evidence. But the status of the notion of cyclicity has constantly been controversial. Importantly, this controversy includes explicitly skeptical views within the Minimalist literature itself on attempts to derive islandhood from phasehood. First we briefly review two such remarks below. This is followed by a critical review of some of the alleged major evidence for EPM.

In a series of papers culminating in his short monograph (Boeckx 2012), Cedric Boeckx argues – building on unpublished work by Markéta Ceplová – that essentially no version of phases will actually wind up defining islands. As an example, Boeckx (2012) considers the attempt by Müller (2010) to derive Huang's (1982) Condition on Extraction Domains from the Phase Impenetrability Condition. The main conclusion of Boeckx is that Müller's attempt fails: a certain set of assumptions about constraints on feature checking and Merge that make crucial reference to the lexical valence list of heads have the unintended consequence that a *wh*-word can escape the boundary created by phase and move to a higher position (see Boeckx 2012, 63–71, and Kubota and Levine 2020, 284–289 for more details).

Den Dikken (2018) arrives at a similar conclusion, from a somewhat different angle. Following the treatment of valuation in Epstein and Seely 2002, den Dikken points out that on that analysis, information about material that is supposedly buried deeply within successive layers of phases must still be retained (i.e., made visible) to the end of the derivation. This leads him to conclude that matrix C should have access to that information, 'which should enable it to attract [a *wh*-word] straight to its specifier, without any intermediate stopovers being necessary along the way' (den Dikken 2018, 65–66). The point here is that the Epstein/Seely formulation embodies an inherent dilemma: the non-local access of information allowed for matrix C would effectively nullify the locality constraint that the very notion of phase/cyclicity is supposed to capture.

If these authors are right, we cannot automatically assume the long-held idea that the notion of phasehood is partly motivated by the explanatory role that it plays in deriving islandhood.[5] This then means that the role that the empirical phenomenon of EPM plays in motivating the theoretical notion of phase and cyclicity is now even bigger than before. This in turn motivates the central goal of the present paper, namely, looking at this notion from a different theoretical angle, one that has logical inference for meaning composition at its core. But before getting to that point, we need to critically review the alleged empirical evidence for cyclicity/EPM, since this empirical literature itself also merits careful scrutiny.

Empirical issues 2.2

There is now a vast literature on reported cases of empirical evidence for EPM. See, for example, van Urk's (2020) recent survey. However,

[5] Also relevant here is the fact that there is now a growing body of literature providing alternative, pragmatic or processing-oriented accounts of many of the classical island constraints. Some important work in this strand of research includes Deane (1992), Kluender (1992, 1998), Hofmeister and Sag (2010) and Chaves and Putnam (2020). See Newmeyer 2016 and Kubota and Levine 2020, Chapter 10 for recent overviews. Even within the Minimalist literature, some authors, such as Dennis Ott, go so far as to claim that islandhood is an 'open wound of syntactic theorizing' (Ott 2014, 290).

upon closer scrutiny, it turns out that there is far more room for dispute than is generally acknowledged. In another recent survey, den Dikken (2018, 69) even goes on to note that '[t]he vast majority of the arguments for successive-cyclic movement available in the literature are based on facts that are at best merely compatible with the hypothesis, not evidence for it'. In this subsection, we review some important counterarguments (some of which seem to have been underestimated) to some of the well-known cases of EPM effects.

2.2.1 EPM effects in French and Chamorro

Among the original group of languages singled out as reflecting EPM effects, French and Chamorro have come in for significant challenge. In the case of French, the acceptability of some of the key examples from Kayne and Pollock (1978) that supposedly demonstrate subject inversion in structurally higher clauses by extraction from a finite embedded clause has been called into question by Bonami *et al.* (1999); according to the latter authors, in such cases only the subject of an embedded clause projected from a head hosting the gap site can undergo this kind of inversion. On the basis of this observation and a wider range of data, Bonami *et al.* argue for an alternative analysis in which the inversion of the subject reflects generalizations about word order rather than sensitivity of an extraction pathway.

In the case of Chamorro, in Chung's (1982) original account, verbs register an agreement pattern with an argument that contains a gap, no matter how deeply embedded. However, even setting aside the theoretical problems (see den Dikken 2017), this account has an empirical flaw: the characterization of the phenomenon by Chung has been argued by Donohue and Maclachlan (1999) to be compatible with an alternative analysis that doesn't rely on the notion of cyclic movement. On the latter authors' view, in what they label 'Philippine-type languages', erosion of a typologically general pattern of voice marking has created the illusion of an exclusive agreement relationship between arguments containing gap sites and the selecting verb.

2.2.2 'Remnant movement' in Afrikaans

The earliest argument for EPM based on partial *wh*-movement, which is essentially a special case of remnant movement, comes from du Plessis (1977), with the paradigm given in (2).

(2) a. Waarvoor werk ons nou eintlik __?
wherefore work we now actually
'For what do we actually work?'

b. Waar werk ons nou eintlik __ **voor**?

c. Waarvoor [dink julle __ [werk ons __]]?
'What do you think we work for?'

d. Waar/wat dink julle [**voor** __ [werk ons __]] ?

(2a) exhibits the more or less default extraction pattern: *waarvoor* appears in Spec,CP with a gap in its presumed argument position. In (2b), however, *waar* has moved, but has left behind the bound form of the preposition with which it is compounded in (2a). (2c) is a long-distance pattern of full *waarvoor* extraction, and (2d) is the crucial case in which *voor* is stranded at an intermediate Spec,CP.

However, as discussed in den Besten (2010), the interpretation of the facts just given appears to be simply mistaken, or at least equivocal (see also den Dikken (2009), who refutes similar arguments for cyclicity in Dutch based on similar sorts of considerations).[6] In particular, den Besten notes that in (2d), the application of the matrix V2 rule in Afrikaans (moving the verb *dink* from the clause-final underlying position immediately before the complement clause to the surface position) makes it difficult to tell whether *voor* actually occupies the embedded Spec,CP position or is an element of the matrix clause syntactically. Since Afrikaans V2 is a root clause constraint, one can observe *voor*'s actual underlying location more accurately using an embedded *wh*-interrogative example:

[6] There is another problem with this remnant movement analysis. As noted by du Plessis himself, the alleged stranded preposition in (2b) and (2d) has to be *voor*, instead of the standard free form preposition *vir* (as in *vir wat* 'for what'). *Voor* is identical in form with the part of the compound *wh*-PP *waarvoor* in (2a), and this form identity is supposedly what motivates du Plessis' analysis via remnant movement. However, outside of this remnant movement literature, there is no known case in which a syntactic operation pries apart a lexical item in the way it does in (2b) and (2d) (on du Plessis' analysis). It is unclear how this violation of the lexical integrity principle (see, e.g., Bresnan and Mchombo 1995; Manning *et al.* 1999) can be accounted for in a cyclic movement-based analysis.

(3) a. Ek sou graag wou weet [$_{CP}$ waar [julle **voor** dink [$_{CP}$ dat [ons werk]]]]
'I would like to know what you think we work for.'
b. *Ek sou graag wou weet [$_{CP}$ waar [julle dink [$_{CP}$ **voor** [dat ons werk]]]]

The contrast in (3) shows that *voor* can end up stranded as a matrix clause element (presumably via clause-internal fronting of the *wh*-element) but cannot occupy an embedded Spec,CP. According to den Besten (2010), du Plessis's (1977) crucial example (2d) should thus be analyzed on a par with (3a) (modulo the V2 word order) rather than the ungrammatical (3b), and hence cannot be taken to involve an intermediate Spec,CP remnant.

2.2.3 *Wh*-copying

The *wh*-copying construction in German (and some other languages) has often been invoked in the literature as evidence for successive cyclicity. This phenomenon is illustrated in (4), where a copy of the *wh*-word appears in overt syntax at an intermediate Spec,CP position:

(4) **Wen** meint Karl [$_{CP}$ **wen** wir __ gewählt haben]?
who thinks Karl who we voted.for have
'Who does Karl think we voted for?'

Den Dikken (2017) notes several issues with an analysis of *wh*-copying in terms of successive cyclic movement. First, as den Dikken notes, prospects for a cyclic analysis start looking murky as soon as we turn our attention to cases involving complex *wh*-phrases.

(5) a. ***Wessen Studenten** denkst du **wessen Studenten** man
whose students think you whose students one
einladen sollte?
invite should
intended: 'Whose students do you think should be invited?'
b. **Wen** denkst du [**wen von den Studenten**] man
who think you who of the students one
einladen sollte?
invite should
'Which of the students do you think should be invited?'

(5a) shows that pronouncing a literal copy of a complex *wh*-phrase at each landing site is ungrammatical. The example improves by replacing one of the two complex *wh*-phrases by a simpler form as in (5b). This is exactly the opposite of what one would expect on the simplest version of 'form-identical multiple copy'-type analysis.

The above paradigm seems already quite troublesome, but den Dikken notes further difficulties for a cyclic movement analysis. Specifically, with respect to scope interpretation, the *wh*-copying phenomenon does not behave like standard overt long-distance movement, but is more similar to the *wh*-scope marking construction (e.g., *Was meint Karl wen wir __ gewählt haben?*, where instead of the *wh*-pronoun *wen*, the *wh*-word at the matrix level is the fixed form *was* 'what'). This and the problem with complex *wh*-phrases leads den Dikken to conclude that the *wh*-copying construction had better be analyzed as a special type of *wh*-scope marking and should not be viewed as a case of long-distance movement with copies in a single derivational chain pronounced at intermediate and final landing sites.

SKETCH OF A NEW ANALYSIS 3

A characteristic that distinguishes our approach from all known formulations of cyclicity in the literature is that it takes the cyclicity effect to be a reflex of the way in which meaning composition interacts with syntax. This is technically implemented via constraints on the forms of logical proofs corresponding to linguistic derivations. The full formal analysis (presented in Section 5) is formulated in a version of Type-Logical Grammar (TLG), whose formal details may feel dauntingly technical to some. However, as explained below, it can essentially be seen as a formalization of the LF-based theory in mainstream syntax. To make the exposition easier to follow, we present the analysis in two steps. This section presents the gist of the analysis in informal terms. This is followed by a compact introduction to TLG in Section 4 and the full formal analysis of EPM in Section 5.

3.1 *Derivations as proofs*

In TLG, linguistic derivations are formally logical proofs. Roughly speaking, Merge (in minimalist terms) corresponds to modus ponens ($P \rightarrow Q, P \models Q$) and Move to hypothetical proof (assuming P, deriving some conclusion Q, and then, drawing the real conclusion $P \rightarrow Q$ by withdrawing the hypothesis P). The following derivation for the relative clause *who Bill criticized* __ illustrates the relevant point:

(6)

$$\dfrac{\text{guy}; \mathbf{guy}; \mathrm{N} \quad \dfrac{\begin{array}{l}\lambda\sigma.\text{who} \bullet \sigma(\epsilon);\\ \lambda P\lambda Q\lambda u.\\ \quad Q(u) \wedge P(u);\\ (\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})\end{array} \quad \textcircled{1}\rightarrow \dfrac{\dfrac{\text{bill}; \mathbf{b}; \mathrm{NP} \quad \dfrac{\text{criticized}; \mathbf{criticized}; \mathrm{VP}/\mathrm{NP} \quad \left[\begin{array}{l}\varphi_0;\\ x; \mathrm{NP}\end{array}\right]^1}{\text{criticized} \bullet \varphi_0; \mathbf{criticized}(x); \mathrm{VP}}/\mathrm{E}}{\text{bill} \bullet \text{criticized} \bullet \varphi_0; \mathbf{criticized}(x)(\mathbf{b}); \mathrm{S}}\backslash\mathrm{E}}{\lambda\varphi_0.\text{bill} \bullet \text{criticized} \bullet \varphi_0; \lambda x.\mathbf{criticized}(x)(\mathbf{b}); \mathrm{S}\upharpoonright\mathrm{NP}}\upharpoonright\mathrm{I}^1}{\text{who} \bullet \text{bill} \bullet \text{criticized} \bullet \epsilon; \lambda Q\lambda u.Q(u) \wedge \mathbf{criticized}(u)(\mathbf{b}); \mathrm{N}\backslash\mathrm{N}}\upharpoonright\mathrm{E}}{\text{guy} \bullet \text{who} \bullet \text{bill} \bullet \text{criticized} \bullet \epsilon; \lambda u.\mathbf{guy}(u) \wedge \mathbf{criticized}(u)(\mathbf{b}); \mathrm{N}}\backslash\mathrm{E}$$

Here, linguistic signs are written as triples of prosodic form, semantics and syntactic category (or 'syntactic type'). The key steps in the derivation in (6) can be paraphrased in prose as follows.

- The NP with prosody φ_0 is a hypothetically assumed NP (the square brackets around it indicate its status as such). With this hypothesis, we derive a complete S corresponding to the body of the relative clause *Bill criticized* __ (immediately above ①).
- The crucial step is the next one (①). At this point, the hypothesis is *withdrawn*, yielding an expression of category S↾NP, a sentence containing an NP-type gap.
- The relative pronoun then takes this gapped sentence as its first argument and returns a backward nominal modifier of type N\N.

The exact way in which prosodic lambda binding in (6) ensures the effect of 'overt movement' of the relative pronoun will be discussed in Section 4, so, we omit the details here. The important point here, which will be crucial in the implementation of EPM, is that hypothetical reasoning (deriving a gapped S↾NP from a *hypothetical* proof of S on the assumption of NP) is the underlying principle that derives the effect of 'movement' (in the standard parlance) and that syntactic/prosodic form and semantics are derived in tandem at each step.

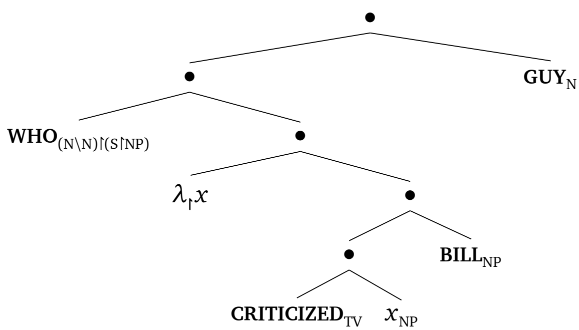
To facilitate the ensuing discussion, we notate the proof trees of the sort in (6) in an alternative, simpler format. Again, we gloss over details radically in this section. All one needs to know at this point is that this alternative notation has solid theoretical underpinnings (explained in detail in Section 4) and that it looks very similar to LF trees of the sort familiar from, e.g., Heim and Kratzer 1998.

We first posit the following constants (written in small capitals) for each of the lexical items used in the derivation in (6) (in what follows, TV is an abbreviation for (NP\S)/NP):

(7) $\textbf{CRITICIZED}_{\mathrm{TV}}$ = criticized; **criticized**; TV
$\textbf{WHO}_{(\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})}$ =
$\lambda\sigma.\mathsf{who} \bullet \sigma(\epsilon); \lambda P\lambda Q\lambda u.Q(u)\wedge P(u); (\mathrm{N}\backslash\mathrm{N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})$
$\textbf{BILL}_{\mathrm{NP}}$ = bill; **b**; NP
$\textbf{GUY}_{\mathrm{N}}$ = guy; **guy**; N

Then, the proof tree in (6) can be rewritten as in (8):

(8)



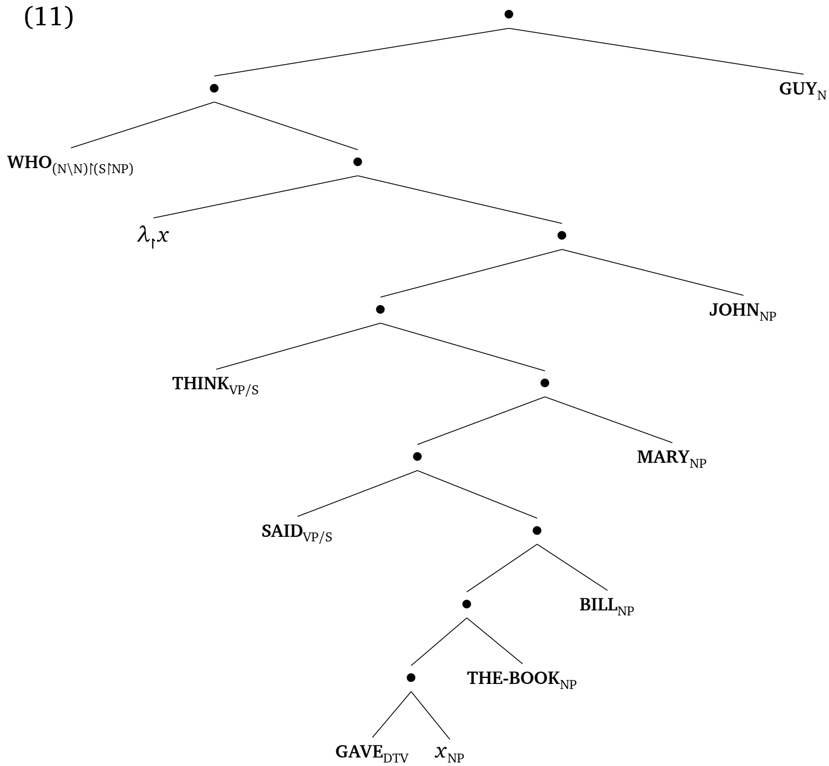
All we have done here is replace the tripartite signs at the leaves by the abbreviations in (7) and write the tree upside down. Thus, from (8) and (7), the original proof in (6) (with more information explicitly written at each node) is fully recoverable.

Note that this way of establishing the relationship between the *wh*-operator and the corresponding hypothesis can cross multiple levels of embedding, since all that's involved is the general mechanism for deducing expressions of type S↾NP, a sentence missing an NP in some arbitrary position inside. Thus, a long-distance relativization example (9) can be analyzed by exactly the same mechanism as in (10)/(11).

(9) the guy who John thinks Mary said Bill gave __ the book

(10) $\textbf{WHO}_{(N\backslash N)\upharpoonright(S\upharpoonright NP)}(\lambda_{\upharpoonright}x.\textbf{THINK}_{VP/S}(\textbf{SAID}_{VP/S}(\textbf{GAVE}_{DTV}(x_{NP})$
$(\textbf{THE-BOOK}_{NP})(\textbf{BILL}_{NP}))(\textbf{MARY}_{NP}))(\textbf{JOHN}_{NP}))(\textbf{GUY}_{N})$

(11)



3.2 *Irish complementizer marking*

We illustrate the analysis with the Irish complementizer choice reported in McCloskey 1979.[7] In this subsection, we review the key data, using our pseudo-language Iringlish from Section 1 for expository convenience. We start with clausal embedding without any extraction. In

[7] As noted by Chaves and Putnam (2020), McCloskey's original proposal in terms of cyclic movement does not seem to be entirely unproblematic in view of the Minimalist theory of movement. In the latter, movement is driven by the need to check uninterpretable features, and in McCloskey 2002, McCloskey himself is essentially forced to posit a number of uninterpretable features which themselves lack independent empirical support.

this case, as shown in (12)–(13), the complementizers (the counterpart of *that* in English) are all realized as *goN*.

(12) I thought **goN** [he would be there].

(13) I said **goN** [I thought **goN** [he would be there]].

As explained in Section 1, when the complementizer position is on an extraction pathway, the alternative form *aL* is used. Thus, for example, in the following (14), the lower clause is marked by *goN*, but the higher clause is marked by *aL*:

(14) the man **aL** [__ thought **goN** [he would be there]]

The examples in (15)–(16), with a multiple chain of *aL* complementation, show that the linkage between the filler and the gap is registered over an arbitrary number of structural levels.

(15) the man **aL** [I thought **aL** [__ would be there]]

(16) the man **aL** [I said **aL** [I thought **aL** [__ would be there]]]

Regardless of the depth of the extraction, as soon as the gap site is identified, all lower clauses which themselves are not associated with an extraction will be marked by *goN*, a point illustrated in (14) and at still greater structural depth in (17).

(17) the man **aL** [he said **aL** [__ thought **goN** [he would be there]]]

Accounting for extraction pathway marking 3.3

The pattern displayed by Iringlish is simple: the form of the complementizer is sensitive to the existence of an unbound gap in the complement clause. But how can we encode this restriction? The apparent dilemma here is that neither cyclic movement nor feature percolation is native to the architecture of TLG. In the analysis of extraction sketched above in Section 3.1, the filler/gap identification is mediated via a single instance of hypothetical reasoning. So, nothing 'moves' literally (let alone in a successive cyclic way), nor is there any structure-manipulation operation or feature percolation of any sort.

The answer comes from seeing proofs as structured objects that linguistic signs can (at least partly) make reference to. Mainstream syntacticians will probably consider this idea more or less unobjectionable (since LF trees are representational objects anyway), but advocates of (traditional) categorial grammar may find it alarming. This is because we need to part with one influential assumption that has dominated CG research over the past several decades. What we need to give up is the idea that the grammar cannot access the internal structures of syntactic proofs.[8] For ardent advocates of direct compositionality, this may appear to be a high price to pay. For such readers, we note that the challenge here is to come up with an explicit analysis of EPM facts in a theory that abides by direct compositionality – a task which, so far as we can tell, is far from trivial.

The proof term notation of derivations introduced above enables a concise formulation of the EPM patterns exhibited by the Iringlish (or Irish) data above. We illustrate this point with a fragment of Iringlish with the lexicon in (18).[9]

(18) a. $\mathbf{WBT}_{\mathrm{NP\backslash S}} = \mathsf{would} \bullet \mathsf{be} \bullet \mathsf{there}; \lambda x.\mathbf{located}(x)(\mathbf{there}); \mathrm{NP\backslash S}$

b. $\mathbf{MAN}_{\mathrm{N}} = \mathsf{man}; \mathbf{man}; \mathrm{N}$

c. $\mathbf{THOUGHT}_{(\mathrm{NP\backslash S)/S'}} = \mathsf{thought}; \mathbf{thought}; (\mathrm{NP\backslash S})/\mathrm{S'}$

[8] While the origin of this idea is unclear, it likely stems from the view in classical Montague Grammar that the translation language is an intermediate step that is in principle eliminable (see, e.g., Dowty *et al.* 1981 and Cooper 1983). It is worth noting in this connection that Dowty (2007), in his later work, has emphasized that compositionality is a *methodological* principle rather than a fixed or fundamental assumption.

[9] For expository convenience, the fragment presented in the main text involves an empty relativizer **REL**. Proponents of lexicalist theories of syntax might find this treatment objectionable. For such readers, we'd like to point out that the effect of **REL** can be lexicalized easily with an alternative, relativized version of *aL* shown below, which can be thought of as a lexicalization of function composition of complementizer *aL* and relativizer **REL** in the underlying calculus, as in (ia):

(i) a. **AL-REL**

$= \lambda_{\upharpoonright} f.\mathbf{REL}(\lambda_{\upharpoonright} x.\mathbf{AL}(f_{\mathrm{S\upharpoonright NP_{+\mathit{wh}}}}(x_{\mathrm{NP_{+\mathit{wh}}}})))$

$= \lambda\sigma.\mathsf{aL} \bullet \sigma(\epsilon); \lambda P \lambda Q \lambda y.Q(y) \wedge P(y); (\mathrm{N\backslash N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP}_{+\mathit{wh}})$

b. ***GON-REL**

$= \lambda_{\upharpoonright} f.\mathbf{REL}(\lambda_{\upharpoonright} x.\mathbf{GON}(f_{\mathrm{S\upharpoonright NP_{+\mathit{wh}}}}(x_{\mathrm{NP_{+\mathit{wh}}}})))$

Note that such a lexicalized variant is unavailable for *goN*: as in (ib), it violates the free-variable prohibition restriction imposed on *goN* in (18f).

d. $\textbf{SAID}_{(NP\backslash S)/S'}$ = said; **said**; $(NP\backslash S)/S'$

e. $\textbf{AL}_{S'/S}$ = aL; $\lambda p.p$; S'/S
   where for any α, $\textbf{AL}(\alpha)$ is defined only if $fv_{X_{+wh}}(\alpha) \neq \emptyset$

f. $\textbf{GON}_{S'/S}$ = goN; $\lambda p.p$; S'/S
   where for any α, $\textbf{GON}(\alpha)$ is defined only if $fv_{X_{+wh}}(\alpha) = \emptyset$

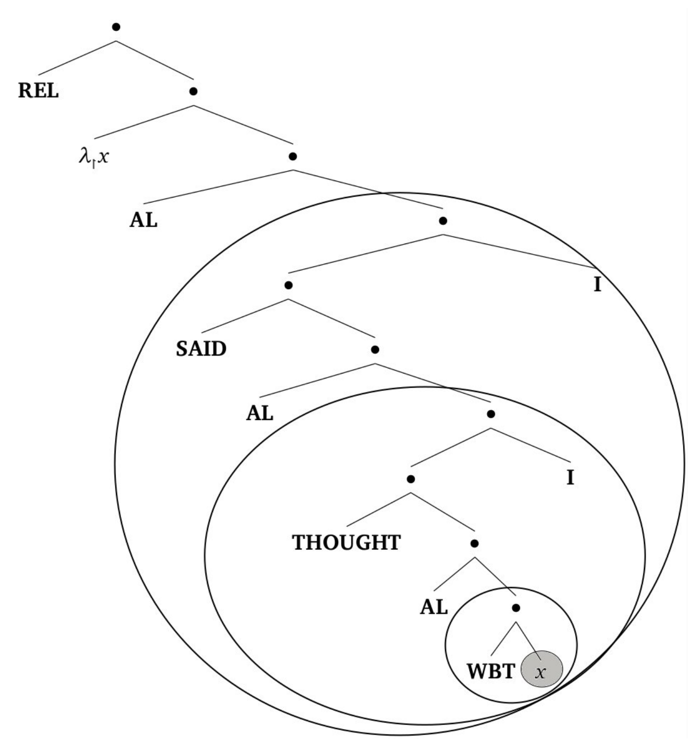
g. $\textbf{REL}_{(N\backslash N)\upharpoonright(S'\upharpoonright NP_{+wh})}$ =
   $\lambda\sigma_2.\sigma_2(\epsilon)$; $\lambda P\lambda Q\lambda y.Q(y) \wedge P(y)$; $(N\backslash N)\upharpoonright(S'\upharpoonright NP_{+wh})$

The key components of this analysis are the restrictions imposed on *aL* and *goN* that refer to the structures of the terms given as their (first) arguments. fv_Φ is the standard, inductively defined function that returns all free variables contained in a term, except that it filters the output of the general purpose *fv* to type Φ. We illustrate with concrete examples below how these lexical constraints on complementizers properly restrict their distributions.

The topmost relative clause in (19) can now be derived as in (20).

(19) the man **aL** [I said **aL** [I thought **aL** [__ would be there]]]

(20)



Here, each token of *aL* applies to a clausal complement containing the free variable x and hence is legal.

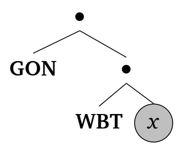
The ungrammaticality of the examples in (21) also follows immediately. In the case of (21a), *goN* is used instead of *aL* in the subproof corresponding to the innermost clause. This violates the constraint $fv_{\mathrm{X}_{+wh}}(\alpha) = \emptyset$ on the first argument of *goN*. Similarly, in (21b), *goN* replaces the first *aL* in the subproof corresponding to the outermost clause. Here again, the relevant 'no unbound $+wh$ hypothesis' constraint on *goN* is violated.

(21) a. *the man **aL** [I said **aL** [I thought **goN** [__ would be there]]]

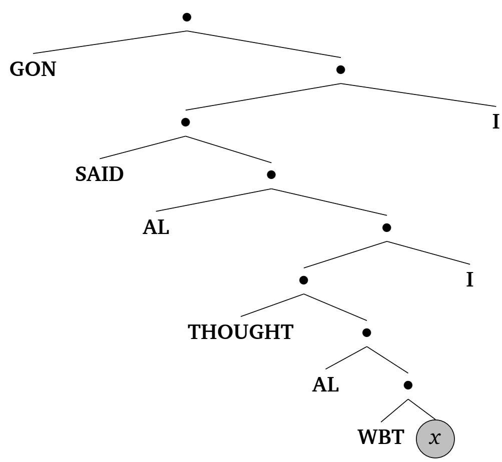
b. *the man **goN** [I said **aL** [I thought **aL** [__ would be there]]]

The offending subterms in the proofs for (21a,b) are shown in (22).

(22) a.



b.



Thus, by making the lexical entries of the complementizers sensitive to the existence of open hypotheses in subproofs, we obtain a simple and straightforward analysis of EPM. Since the existence of open hypotheses conceptually corresponds to the fact that the complementizer is licensed at a point in the derivation at which filler-gap linkage is not yet established, we obtain the effect of 'cyclicity' without literally encoding a structure-manipulation operation of cyclic movement.

Some remarks are in order regarding the possible similarities and differences between the present analysis of EPM as 'proof structure making' and the more standard configurational approach in derivational syntax (see Citko 2014 for an overview of the latter). The similarity should be clear. In both approaches, linguistic derivations are regarded as structured objects and the grammar offers one way or another for making reference to part of the 'derivational history' that certain lexical items (or other aspects of grammar) are sensitive to.

Turning to differences, we see at least two aspects in which our proposal substantially differs from the standard view. First, by viewing EPM as a mere reflection of the 'hypothesis containing' status of a subproof, our approach predicts that 'phase boundaries' are not necessarily limited to a small set of categories (standardly, CP and vP).[10] This is perhaps the single most important difference. What constitutes the exact set of 'phase boundaries' is itself a controversial issue in Minimalist syntax (see Legate 1998 and especially Matushansky 2005 for some discussion on this thorny issue), and we are not prepared to get into an in-depth discussion on this topic, but one point is worth noting: in Minimalist formulations, there has to be some conceptual basis for restricting the set of 'phase boundaries', and it has sometimes been suggested that this may come from semantic considerations, with CP corresponding to a proposition-denoting unit (cf., e.g., Chomsky 2000; Hinzen 2012). If such a semantic characterization of 'phase boundaries' is tenable, that would be entirely compatible with our account, since in TLG, there is a tight correspondence between syntactic types and semantic types, and at each step of derivation, the full denotation of the linguistic expression being derived is available.

This then relates to the second major difference. In the standard phase-based approach, the correspondence between syntactic computation and compositional semantics is somewhat unclear. It is only via the explicit structural operation of movement (or external merge and

[10] In Section 6, we offer brief speculations on how one might go about making sense of what seems like a skewed syntactic distribution of EPM items cross-linguistically under the meaning-centered approach that our proof-theoretic perspective embodies.

the specific way in which two copies of the same lexical item get interpreted at the CI component) that we get the effect of variable binding. Our approach captures the connection between 'movement' and 'variable binding' more straightforwardly, since 'movement' is by definition nothing other than variable binding (or hypothetical reasoning) in the underlying logic governing the correspondence between surface form and the compositional meaning. The analysis of EPM crucially exploits this property of the TLG architecture (and the formal tools available in it for formulating meta-statements pertaining to the statuses of subproofs), a point we get back to at the end of the paper.

Our approach essentially embodies a *meaning-centered* perspective on EPM. We believe that this represents at least an interesting enough alternative to the standard structure-driven approach. It may appear to have some glaring loose ends, but we believe that the conceptual simplicity is attractive enough to compensate for this possible shortcoming (which after all relates to a still open and controversial issue).

4 LONG-DISTANCE DEPENDENCIES IN HYBRID TLG

This section is meant to serve two purposes: to introduce Hybrid TLG as a syntactic framework and to illustrate its workings with an analysis of pied-piping in relative clauses. The choice of the empirical phenomenon is motivated by the fact that pied-piping exhibits properties of both 'overt' and 'covert' movement in derivational syntax. A recasting of the movement-based analysis of pied-piping from mainstream generative syntax in Hybrid TLG – building on an earlier analysis by Morrill (1994) – illustrates clearly the way in which TLG handles complex mapping between form and meaning. There is already substantial literature on linguistic applications of TLG (see, e.g., Morrill 1994; Carpenter 1997; Kubota and Levine 2020), and readers are encouraged to refer to these sources for more information about TLG as a syntactic framework. Handbook articles such as Moortgat 2011, 2014 and Kubota 2021 are also useful sources of reference.

The full system of Hybrid TLG comprises three logical connectives $/$, $\backslash$ and $\upharpoonright$, and has Elimination and Introduction rules for all these.

However, since the linguistic phenomena we deal with in this paper do not involve hypothetical reasoning with the directional slashes / and \, our presentation below focuses on the way in which the directional slashes / and \ are used for licensing local function-argument structures and on the use of the ↾ connective for modeling 'movement' operations (this corresponds to the system introduced in Section 2.3 of Kubota and Levine 2020). The more complex Introduction rules for / and \ are discussed only briefly in Section 4.4.

AB grammar 4.1

We start with a simple fragment called the *AB grammar* (Ajdukiewicz 1935; Bar-Hillel 1953), consisting of just the two syntactic rules in (23):

(23) a. Forward Slash Elimination

$$\frac{b; B \quad a; A/B}{a \bullet b; A}/\mathrm{E}$$

b. Backward Slash Elimination

$$\frac{b; B \quad a; B\backslash A}{b \bullet a; A}\backslash\mathrm{E}$$

With the somewhat minimal lexicon in (24), we can license a simple transitive verb sentence (25) as in (26). The two slashes / and \ are used to form complex syntactic categories, or syntactic types, indicating valence information: The transitive verb *loves* is assigned the syntactic type (NP\S)/NP since it first combines with an NP to its right (i.e. the direct object) and then another NP to its left (i.e. the subject).

(24) a. john; NP
b. mary; NP
c. ran; NP\S
d. loves; (NP\S)/NP

(25) John loves Mary.

(26)

$$\frac{\text{john; NP} \quad \dfrac{\text{mary; NP} \quad \text{loves; (NP\S)/NP}}{\text{loves} \bullet \text{mary; NP\S}}/\mathrm{E}}{\text{john} \bullet \text{loves} \bullet \text{mary; S}}\backslash\mathrm{E}$$

There is one thing to keep in mind about proof notation. In the presentation of proofs and rules adopted in (23) and (26), the word order is reflected solely in the prosodic annotations at each node of the tree, and the left and right order of the premises in a subtree does not have anything to do with the surface word order of English sentences (in the rest of the paper, we generally align the order of premises with the actual word order, but this is only for expository ease).

Syntactic types are defined recursively. For the AB grammar, this can be concisely written using the so-called 'BNF notation' as follows (the exact choice of the set of basic types is an empirical question):

(27) $\mathscr{A} := \{ \text{S, NP, N, PP, } \ldots \}$ (atomic type)
$\mathscr{T} := \mathscr{A} \mid \mathscr{T}\backslash\mathscr{T} \mid \mathscr{T}/\mathscr{T}$ (type)

In words, anything that is an atomic type is a type, and any complex expression of form $A\backslash B$ or A/B where A and B are both types is a type.

As should already be clear in the above illustration, categorial grammar lexicalizes the valence (or subcategorization) properties of linguistic expressions, and this is transparently represented in the syntactic types of functional expressions (such as verb lexical entries). Here are some more sample lexical entries:

(28) a. ran; NP\S
b. read; (NP\S)/NP
c. introduces; (NP\S)/PP/NP

4.2 *Syntax-semantics interface*

Assuming the standard recursive definition of semantic types as in (29) (with basic types e (individuals) and t (truth values) for an extensional fragment), we can define the function Sem that returns, for each syntactic type given as input, its semantic type, as in (30)–(31).

(29) a. $\mathscr{A}_\sigma := \{ e, t \}$ (atomic semantic type)
b. $\mathscr{T}_\sigma := \mathscr{A}_\sigma \mid \mathscr{T}_\sigma \rightarrow \mathscr{T}_\sigma$ (semantic type)

(30) (Base Case)
a. $\mathsf{Sem}(\text{NP}) = \mathsf{Sem}(\text{PP}) = e$
b. $\mathsf{Sem}(\text{N}) = e \rightarrow t$
c. $\mathsf{Sem}(\text{S}) = t$

(31) (Recursive Clause)
For any complex syntactic type of the form A/B (or $B\backslash A$),
$\mathsf{Sem}(A/B)\ (= \mathsf{Sem}(B\backslash A)) = \mathsf{Sem}(B) \rightarrow \mathsf{Sem}(A)$

For example, assuming that VP adverbs such as *quickly* are of type (NP\S)\(NP\S), we can determine their semantic type based on the syntactic type by following the definitions in (29)–(31):

(32) $\text{Sem}((\text{NP}\backslash\text{S})\backslash(\text{NP}\backslash\text{S}))$
$= \text{Sem}(\text{NP}\backslash\text{S}) \to \text{Sem}(\text{NP}\backslash\text{S})$
$= (\text{Sem}(\text{NP}) \to \text{Sem}(\text{S})) \to (\text{Sem}(\text{NP}) \to \text{Sem}(\text{S}))$
$= (e \to t) \to (e \to t)$

In other words, the syntactic type (NP\S)\(NP\S) transparently represents the semantic type of a VP modifier as an $e \to t$ property modifier.

Syntactic rules with semantics can then be written as in (33) (where the semantic effect of these rules is *function application*) and a sample derivation with semantic annotation is given in (34).

(33) a. Forward Slash Elimination

$$\frac{a;\ \mathscr{F};\ A/B \quad b;\ \mathscr{G};\ B}{a \bullet b;\ \mathscr{F}(\mathscr{G});\ A}\ /\text{E}$$

b. Backward Slash Elimination

$$\frac{b;\ \mathscr{G};\ B \quad a;\ \mathscr{F};\ B\backslash A}{b \bullet a;\ \mathscr{F}(\mathscr{G});\ A}\ \backslash\text{E}$$

(34)

$$\dfrac{\text{john};\ \mathbf{j};\ \text{NP} \quad \dfrac{\dfrac{\text{chased};\ \mathbf{chased};\ (\text{NP}\backslash\text{S})/\text{NP} \quad \text{mary};\ \mathbf{m};\ \text{NP}}{\text{chased} \bullet \text{mary};\ \mathbf{chased}(\mathbf{m});\ \text{NP}\backslash\text{S}}\ /\text{E} \quad \text{patiently};\ \mathbf{patiently};\ (\text{NP}\backslash\text{S})\backslash(\text{NP}\backslash\text{S})}{\text{chased} \bullet \text{mary} \bullet \text{patiently};\ \mathbf{patiently}(\mathbf{chased}(\mathbf{m}));\ \text{NP}\backslash\text{S}}\ \backslash\text{E}}{\text{john} \bullet \text{chased} \bullet \text{mary} \bullet \text{patiently};\ \mathbf{patiently}(\mathbf{chased}(\mathbf{m}))(\mathbf{j});\ \text{S}}\ \backslash\text{E}$$

4.3 *Adding the vertical slash for 'movement'*

The AB grammar introduced above deals with local licensing of arguments via the Elimination rules for / and \. This roughly corresponds to simple phrase structure grammar (or context-free grammar) without 'movement' operations. In order to model phenomena that involve both 'covert' and 'overt' movement (in the derivational terminology), we need to extend the underlying logic. In Hybrid TLG, this is done by introducing functional expressions in the prosodic representations of linguistic signs written as λ-terms (Oehrle 1994; de Groote 2001; Muskens 2003; Mihaliček and Pollard 2012). As will become clear below, λ-binding of variables in the prosodic representations makes it possible to 'reason about' linguistic expressions in which something is missing in the middle. This technique is crucially exploited in the analysis of relative clauses in (38) and (40) below.

Building on this tradition, we introduce into our system a new connective $\upharpoonright$ called the *vertical slash*, for order-insensitive mode of implication (as with /, we write the argument to the right for $\upharpoonright$). For this connective, we posit the following two rules:

(35) a. Vertical Slash Introduction

$$\cfrac{\begin{array}{ccc} \vdots & [\varphi; x; A]^n & \vdots \\ \hline \vdots & \vdots & \vdots \\ \hline \multicolumn{3}{c}{b; \mathscr{F}; B} \end{array}}{\lambda\varphi.b; \lambda x.\mathscr{F}; B\upharpoonright A}\upharpoonright\mathrm{I}^n$$

b. Vertical Slash Elimination

$$\frac{a; \mathscr{F}; A\upharpoonright B \quad b; \mathscr{G}; B}{a(b); \mathscr{F}(\mathscr{G}); A}\upharpoonright\mathrm{E}$$

Of these two rules, Vertical Slash Elimination (35b) is simpler. It licenses a structure in which a linguistic expression that has functional prosody (reflected in the syntactic type $A\upharpoonright B$) combines with its argument (of syntactic type B). The rule specifies that in such function-argument pairs (i.e., $A\upharpoonright B$ and B), the two items are combined by function application in both semantics and prosody.

The workings of the Vertical Slash Introduction rule (35a) is somewhat more complex, but the underlying idea is simple. This rule licenses a type of proof in which some linguistic expression (the bracketed expression with index n) is hypothetically assumed to derive an intermediate conclusion (on the penultimate line with type B). The rule then licenses an expression of type $B\upharpoonright A$ by withdrawing the hypothesis A. The corresponding effect in the semantic and prosodic components is λ-binding of the variables introduced by the hypothesis A. The semantic λ-binding should make obvious sense (given the analogy to movement). What's novel (for those unfamiliar with the subspecies of CG stemming from Oehrle 1994) is the λ-binding in the prosodic component. This will be illustrated with an example below in (38). The correspondence between a hypothesis and the $\upharpoonright$I step at which it is withdrawn in the proof tree is kept track of by the index n, since there may be multiple such pairs within a single proof.

The way this extended system works can be best illustrated by concrete examples, so let us now examine a simple analysis of English relative clauses. The key idea is that the new rules just introduced enable us to 'reason about' linguistic expressions in which some material is missing. For example, in (36), the body of the relative clause *Bill criticized* __ is analyzed as S$\upharpoonright$NP, a sentence missing an NP.

(36) the guy who Bill criticized __

We posit the following entry for the relative pronoun *who* in which both the semantics and the prosody are higher-order functions.

(37) $\lambda\sigma.\text{who} \bullet \sigma(\epsilon);\ \lambda P \lambda Q \lambda u.Q(u) \wedge P(u);\ (\text{N}\backslash\text{N})\upharpoonright(\text{S}\upharpoonright\text{NP})$

We can then license (38) for (36) (the dotted lines in (38) just show the β-reduction steps for the prosodic term, and are not part of the syntactic derivation; in what follows, VP is an abbreviation for NP\S).

(38)

$$\text{criticized};\ \textbf{criticized};\ \text{VP/NP} \qquad [\varphi_0;\ x;\ \text{NP}]^1$$
$$\text{bill};\ \textbf{b};\ \text{NP} \qquad \overline{\text{criticized} \bullet \varphi_0;\ \textbf{criticized}(x);\ \text{VP}}\ /\text{E}$$
$$\overline{\text{bill} \bullet \text{criticized} \bullet \varphi_0;\ \textbf{criticized}(x)(\textbf{b});\ \text{S}}\ \backslash\text{E}$$
$$\text{①}\!\to \overline{\lambda\varphi_0.\text{bill} \bullet \text{criticized} \bullet \varphi_0;\ \lambda x.\textbf{criticized}(x)(\textbf{b});\ \text{S}\upharpoonright\text{NP}}\ \upharpoonright\text{I}^1$$
$$\lambda\sigma.\text{who} \bullet \sigma(\epsilon);\ \lambda P \lambda Q \lambda u.Q(u) \wedge P(u);\ (\text{N}\backslash\text{N})\upharpoonright(\text{S}\upharpoonright\text{NP})$$
$$\overline{\lambda\sigma[\text{who} \bullet \sigma(\epsilon)](\lambda\varphi_0.\text{bill} \bullet \text{criticized} \bullet \varphi_0);\ \lambda Q \lambda u.Q(u) \wedge \textbf{criticized}(u)(\textbf{b});\ \text{N}\backslash\text{N}}\ \upharpoonright\text{E}$$
$$\cdots\ \text{who} \bullet \lambda\varphi_0[\text{bill} \bullet \text{criticized} \bullet \varphi_0](\epsilon);\ \lambda Q \lambda u.Q(u) \wedge \textbf{criticized}(u)(\textbf{b});\ \text{N}\backslash\text{N}$$
$$\cdots\ \text{who} \bullet \text{bill} \bullet \text{criticized} \bullet \epsilon;\ \lambda Q \lambda u.Q(u) \wedge \textbf{criticized}(u)(\textbf{b});\ \text{N}\backslash\text{N}$$
$$\text{guy};\ \textbf{guy};\ \text{N}$$
$$\overline{\text{guy} \bullet \text{who} \bullet \text{bill} \bullet \text{criticized} \bullet \epsilon;\ \lambda u.\textbf{guy}(u) \wedge \textbf{criticized}(u)(\textbf{b});\ \text{N}}\ \backslash\text{E}$$

The derivation in (38) can be paraphrased in prose as follows.

- The NP with prosody φ_0 is a hypothetically assumed NP (the square brackets around it indicate its status as such). With this hypothesis, we derive a complete S corresponding to the body of the relative clause *Bill criticized* __ (immediately above ①).
- The crucial step is the next one (①). At this point, the hypothesis is *withdrawn* with the ↾-Introduction rule. This yields an S↾NP, a sentence containing an NP-type gap. The string position of the gap is kept track of by λ-binding the prosodic variable φ_0.
- The relative pronoun, with the lexical specification in (37), then takes this gapped sentence as its first argument and returns a backward nominal modifier of type N\N. (Semantically, the relative pronoun denotes an intersective modifier of two properties.)

The final step where the relative pronoun takes a gapped sentence as argument perhaps requires some comment. The key point here is that the prosodic specification of the relative pronoun in (37) is a higher-order function that combines strings in a particular way. Specifically, its first argument σ is the gapped sentence (itself a function of type

st → **st**, that is, a function that maps a string to another string). It feeds an empty string ϵ to σ, thereby filling in the embedded gap position, and concatenates the string who with the string thus obtained. For the purpose of exposition, the relevant β-reduction steps are explicitly shown in the dotted line part in (38).

An important property of this analysis is that the gap can be deeply embedded inside the relative clause. Hypothetical reasoning with the vertical slash works exactly in the same way in the simple example above in which the gap corresponds to a local argument position and in the more complex example in (39) in which the gap is located in an embedded clause with multiple levels of embedding.

(39) the guy who John thinks Mary said Bill gave __ the book

The derivation for (39) is shown in (40).

(40)

$$\dfrac{\text{gave}; \mathbf{gave}; VP/NP/NP \qquad \left[\varphi_0; x; NP\right]^1}{\text{gave} \bullet \varphi_0; \mathbf{gave}(x); VP/NP}\,/E$$

$$\dfrac{\text{gave} \bullet \varphi_0; \mathbf{gave}(x); VP/NP \qquad \text{the} \bullet \text{book}; \mathbf{the\text{-}book}; NP}{\text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{gave}(x)(\mathbf{the\text{-}book}); VP}\,/E$$

$$\dfrac{\text{bill}; \mathbf{b}; NP \qquad \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{gave}(x)(\mathbf{the\text{-}book}); VP}{\text{bill} \bullet \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{gave}(x)(\mathbf{the\text{-}book})(\mathbf{b}); S}\,\backslash E$$

$$\dfrac{\text{said}; \mathbf{said}; VP/S \qquad \text{bill} \bullet \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{gave}(x)(\mathbf{the\text{-}book})(\mathbf{b}); S}{\text{said} \bullet \text{bill} \bullet \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{said}(\mathbf{gave}(x)(\mathbf{the\text{-}book})(\mathbf{b})); VP}\,/E$$

$$\dfrac{\text{mary}; \mathbf{m}; NP \qquad \text{said} \bullet \text{bill} \bullet \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{said}(\mathbf{gave}(x)(\mathbf{the\text{-}book})(\mathbf{b})); VP}{\text{mary} \bullet \text{said} \bullet \text{bill} \bullet \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{said}(\mathbf{gave}(x)(\mathbf{the\text{-}book})(\mathbf{b}))(\mathbf{m}); S}\,\backslash E$$

$$\dfrac{\text{thinks}; \mathbf{think}; VP/S \qquad \text{mary} \bullet \text{said} \bullet \text{bill} \bullet \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{said}(\mathbf{gave}(x)(\mathbf{the\text{-}book})(\mathbf{b}))(\mathbf{m}); S}{\text{thinks} \bullet \text{mary} \bullet \text{said} \bullet \text{bill} \bullet \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{think}(\mathbf{said}(\mathbf{gave}(x)(\mathbf{the\text{-}book})(\mathbf{b}))(\mathbf{m})); VP}\,/E$$

$$\dfrac{\text{john}; \mathbf{j}; NP \qquad \text{thinks} \bullet \text{mary} \bullet \text{said} \bullet \text{bill} \bullet \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{think}(\mathbf{said}(\mathbf{gave}(x)(\mathbf{the\text{-}book})(\mathbf{b}))(\mathbf{m})); VP}{\text{john} \bullet \text{thinks} \bullet \text{mary} \bullet \text{said} \bullet \text{bill} \bullet \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{think}(\mathbf{said}(\mathbf{gave}(x)(\mathbf{the\text{-}book})(\mathbf{b}))(\mathbf{m}))(\mathbf{j}); S}\,\backslash E$$

$$\dfrac{\text{john} \bullet \text{thinks} \bullet \text{mary} \bullet \text{said} \bullet \text{bill} \bullet \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \mathbf{think}(\mathbf{said}(\mathbf{gave}(x)(\mathbf{the\text{-}book})(\mathbf{b}))(\mathbf{m}))(\mathbf{j}); S}{\lambda\varphi_0.\text{john} \bullet \text{thinks} \bullet \text{mary} \bullet \text{said} \bullet \text{bill} \bullet \text{gave} \bullet \varphi_0 \bullet \text{the} \bullet \text{book}; \lambda x.\mathbf{think}(\mathbf{said}(\mathbf{gave}(x)(\mathbf{the\text{-}book})(\mathbf{b}))(\mathbf{m}))(\mathbf{j}); S{\upharpoonright}NP}\,{\upharpoonright}I^1$$

The addition of a new connective ↾ necessitates a revision of the definition of syntactic types and the mapping from syntactic to semantic types. In addition, the grammar now recognizes not just simple strings (of type **st**) but also functions that compose such strings in particular ways as admissible prosodic representations of linguistic expressions. We therefore need to define the mapping from syntactic types to prosodic types as well. The new definitions are in (41)–(45).

Syntactic types:

(41) $\mathscr{A} := \{ \text{S}, \text{NP}, \text{N}, \ldots \}$ (atomic type)

$\mathscr{D} := \mathscr{A} \mid \mathscr{D}\backslash\mathscr{D} \mid \mathscr{D}/\mathscr{D}$ (directional type)

$\mathscr{T} := \mathscr{D} \mid \mathscr{T}\upharpoonright\mathscr{T}$ (type)

Semantic types:

(42) (Base Case)

a. $\mathsf{Sem}(\text{NP}) = \mathsf{Sem}(\text{PP}) = e$

b. $\mathsf{Sem}(\text{N}) = e \to t$

c. $\mathsf{Sem}(\text{S}) = t$

(43) (Recursive Clause)

For any complex syntactic type of the form A/B (or $B\backslash A$, $A\upharpoonright B$), $\mathsf{Sem}(A/B)$ $(= \mathsf{Sem}(B\backslash A) = \mathsf{Sem}(A\upharpoonright B)) = \mathsf{Sem}(B) \to \mathsf{Sem}(A)$

Prosodic types:

(44) (Base Case)

For any directional type $\mathscr{D}$, $\mathsf{Pros}(\mathscr{D}) = \textbf{st}$ (with **st** for 'strings').

(45) (Recursive Clause)

For any complex syntactic type $A\upharpoonright B$ involving $\upharpoonright$,

$\mathsf{Pros}(A\upharpoonright B) = \mathsf{Pros}(B) \to \mathsf{Pros}(A)$.

Note that $\mathscr{D}$ in (41) replaces $\mathscr{T}$ in the earlier definition of syntactic types in (27). The set of syntactic types $\mathscr{T}$ is defined on top of the set of directional types $\mathscr{D}$ (i.e., the complete set of syntactic types in the earlier definition) as in the final clause in (41). This ensures that a vertical slash cannot occur under a directional slash. Thus, S/(S$\upharpoonright$NP) is not a well-formed syntactic type. One way to make sense of this is to think of it as a 'filter' on uninterpretable prosodic objects. An expression of type X/(Y$\upharpoonright$Z) would have to concatenate a string to the left of a function of type **st** $\to$ **st**, but that doesn't make sense.

As the asymmetry between (43) and (45) should make clear, the three slashes /, \ and $\upharpoonright$ are all functional in the semantic domain, but only $\upharpoonright$ is functional in the prosodic domain. This asymmetry corresponds to the fact that lambda binding is involved in the prosody only for the Introduction rule for $\upharpoonright$ (see Section 4.4 for / and \).

Hypothetical reasoning with the directional slashes 4.4

The key notion involved in the analysis of English relative clauses above is hypothetical reasoning, which is essentially a theoretical machinery for 'reasoning about' complex linguistic expressions in which some material is missing from where it is supposed to appear given the specific lexical specifications of items which make up the complex expressions. In the full version of Hybrid TLG, hypothetical reasoning is generalized to the directional slashes / and \ as well. For the sake of completeness, we show the Introduction rules for / and \, and briefly discuss linguistic applications of these rules.

The Slash Introduction rules for / and \ are formulated as in (46).

(46) a. Forward Slash Introduction

$$\dfrac{\dfrac{\begin{array}{ccc} \vdots & \underline{[\varphi; x; A]^n} & \vdots \\ \vdots & \vdots & \vdots \end{array}}{b \bullet \varphi; \mathscr{F}; B}}{b; \lambda x.\mathscr{F}; B/A}{/}\mathrm{I}^n$$

b. Backward Slash Introduction

$$\dfrac{\dfrac{\begin{array}{ccc} \vdots & \underline{[\varphi; x; A]^n} & \vdots \\ \vdots & \vdots & \vdots \end{array}}{\varphi \bullet b; \mathscr{F}; B}}{b; \lambda x.\mathscr{F}; A\backslash B}{\backslash}\mathrm{I}^n$$

The difference between the Introduction rule for the vertical slash introduced above in (35a) and these rules is that in (46), the prosodic variable φ for the hypothesis is simply thrown away (instead of being λ-bound). The position of the missing expression is instead recorded in the forward vs. backward slash distinction in the syntactic type.

This is useful when one wants to assign a directional slash type for some string of words in which some material is missing at the periphery, instead of analyzing such expressions with functional prosodic types. For example, for the string *John loves* in the Right-node Raising example in (47), we want to assign the type S/NP so that it is directly conjoinable with another string *Bill hates* of the same type.

(47) $[_{\mathrm{S/NP}}$ John loves], and $[_{\mathrm{S/NP}}$ Bill hates], $[_{\mathrm{NP}}$ Mary].

The derivation for the string *John loves* in type S/NP is shown in (48).

(48)

$$\textcircled{1}\rightarrow \dfrac{\dfrac{\text{john}; \mathbf{j}; \mathrm{NP} \quad \dfrac{\text{loves}; \mathbf{love}; (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP} \quad [\varphi; x; \mathrm{NP}]^1}{\text{loves} \bullet \varphi; \mathbf{love}(x); \mathrm{NP}\backslash\mathrm{S}}{/}\mathrm{E}}{\text{john} \bullet \text{loves} \bullet \varphi; \mathbf{love}(x)(\mathbf{j}); \mathrm{S}}{\backslash}\mathrm{E}}{\text{john} \bullet \text{loves}; \lambda x.\mathbf{love}(x)(\mathbf{j}); \mathrm{S/NP}}{/}\mathrm{I}^1$$

In prose:

- A complete sentence is formed with the hypothetical NP indexed 1. (This much is the same as in the earlier (38).)
- At the next step (①), the hypothesis is withdrawn just as in (38), but here the string variable φ is thrown away, and the derived

type is S/NP (with type **st** prosody). It is this syntactic type that tells us that this is a sentence missing an NP on the right.

Proof term notation of derivations 4.5

To facilitate the ensuing discussion, we introduce here an alternative notation of derivations, one in which a derivation/proof can be written as a single formal object, specifically a lambda term. This corresponds to Abstract Syntax in Abstract Categorial Grammar (de Groote 2001). It exploits the theoretical result in TLG research building on the so-called Curry-Howard Isomorphism (Howard 1969), which states that there is a one-to-one correspondence between proofs and lambda terms in a simply typed lambda calculus. Essentially, an Elimination step (in natural deduction) in a proof corresponds to function application in the lambda calculus and an Introduction step corresponds to lambda abstraction. With Hybrid TLG, this lambda calculus for writing syntactic proofs needs to be extended to distinguish three types of function application ($\mathrm{app}_/$, $\mathrm{app}_\backslash$, and $\mathrm{app}_\upharpoonright$), and three types of lambda abstraction ($\lambda_/$, $\lambda_\backslash$, and $\lambda_\upharpoonright$), corresponding to the three slashes.[11]

As an illustration, consider the derivation (49) (= (38) above) for a simple relative clause from the previous section.

(49)

$$\dfrac{\begin{array}{c}\text{guy};\\ \mathbf{guy};\\ \mathrm{N}\end{array} \quad \dfrac{\begin{array}{c}\lambda\sigma.\text{who}\bullet\sigma(\epsilon);\\ \lambda P\lambda Q\lambda u.\\ Q(u)\wedge P(u);\\ (\mathrm{N\backslash N})\upharpoonright(\mathrm{S}\upharpoonright\mathrm{NP})\end{array} \quad \dfrac{\dfrac{\begin{array}{c}\text{bill};\\ \mathbf{b};\\ \mathrm{NP}\end{array} \quad \dfrac{\begin{array}{c}\text{criticized};\\ \mathbf{criticized};\mathrm{VP/NP}\end{array} \quad \begin{bmatrix}\varphi_0;\\ x;\mathrm{NP}\end{bmatrix}^1}{\begin{array}{c}\text{criticized}\bullet\varphi_0;\\ \mathbf{criticized}(x);\mathrm{VP}\end{array}}/\mathrm{E}}{\begin{array}{c}\text{bill}\bullet\text{criticized}\bullet\varphi_0;\\ \mathbf{criticized}(x)(\mathbf{b});\mathrm{S}\end{array}}\backslash\mathrm{E}}{\begin{array}{c}\lambda\varphi_0.\text{bill}\bullet\text{criticized}\bullet\varphi_0;\\ \lambda x.\mathbf{criticized}(x)(\mathbf{b});\mathrm{S}\upharpoonright\mathrm{NP}\end{array}}\upharpoonright\mathrm{I}^1}{\text{who}\bullet\text{bill}\bullet\text{criticized}\bullet\epsilon;\ \lambda Q\lambda u.Q(u)\wedge\mathbf{criticized}(u)(\mathbf{b});\ \mathrm{N\backslash N}}\upharpoonright\mathrm{E}}{\text{guy}\bullet\text{who}\bullet\text{bill}\bullet\text{criticized}\bullet\epsilon;\ \lambda u.\mathbf{guy}(u)\wedge\mathbf{criticized}(u)(\mathbf{b});\ \mathrm{N}}\backslash\mathrm{E}$$

We use the same abbreviation of tripartite linguistic signs in the lexicon introduced in Section 2 (= (7)):

---

[11] This lambda calculus can be thought of as an extension of the bidirectional lambda calculus for the Lambek calculus proposed by Buszkowski (1987) and Wansing (1992). Studying the formal properties of this lambda calculus is an interesting topic on its own, but we leave this task for another occasion.

(50) $\textbf{CRITICIZED}_{\text{TV}} = \textsf{criticized}; \textbf{past}(\textbf{criticize}); \text{TV}$
$\textbf{WHO}_{(\text{N}\backslash\text{N})\upharpoonright(\text{S}\upharpoonright\text{NP})} =$
$\lambda\sigma.\textsf{who} \bullet \sigma(\epsilon); \lambda P \lambda Q \lambda u.Q(u) \wedge P(u); (\text{N}\backslash\text{N})\upharpoonright(\text{S}\upharpoonright\text{NP})$
$\textbf{BILL}_{\text{NP}} = \textsf{bill}; \textbf{b}; \text{NP}$
$\textbf{GUY}_{\text{N}} = \textsf{guy}; \textbf{guy}; \text{N}$

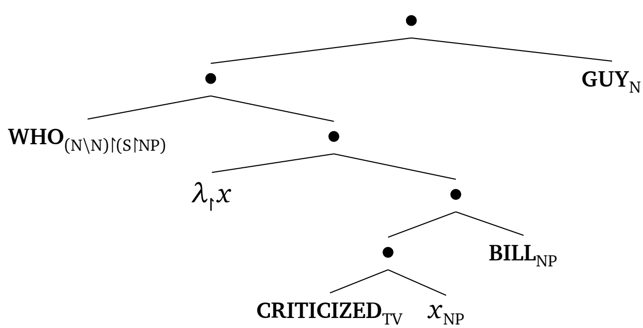
Then, by replacing Slash Elimination by function application and Slash Introduction by lambda abstraction in (49), we obtain the following lambda term, whose syntactic form is isomorphic (i.e., stands in a one-to-one relation) to the natural deduction proof in (49) (the variety of application rule is omitted, since this information is unambiguously recoverable from the syntactic type of the function):

(51) $\textbf{WHO}_{(\text{N}\backslash\text{N})\upharpoonright(\text{S}\upharpoonright\text{NP})}(\lambda_{\upharpoonright}x.\textbf{CRITICIZED}_{\text{TV}}(x_{\text{NP}})(\textbf{BILL}_{\text{NP}}))(\textbf{GUY}_{\text{N}})$

In effect, (51) displays the entire proof narrative exhibited in (49) as a single object: the function corresponding to *criticized* is saturated, with its variable argument undergoing abstraction, yielding an eligible argument for the relative pronoun *who*. Note here that the variable x_{NP} in (51) is a variable in the syntactic logic and is thus formally unrelated to the x in the semantic component of the hypothesis in (49); we use the same variable letter only for expository convenience.

To make it clear that (51) represents underlying semantic composition, and to enhance readability, here is an alternative notation for (51) in the form of a binary tree (already introduced in Section 2):

(52)



Readers familiar with derivational approaches to syntax will recognize a clear resemblance to LF structure. The correspondence to the natural deduction proof tree in (49) should also be easier to see in this format.

The proof term notation is a compact representation of derivations that shows the underlying combinatorics transparently. As we demonstrate below with pied-piping, this is useful in the analyses of complex empirical phenomena involving hypothetical reasoning with the vertical slash (roughly corresponding to 'syntactic movement').

Pied-piping as 'overt and covert' movement 4.6

In the analysis of English relative clauses above, the semantic and syntactic linkage between the extracted material, the relative pronoun and the rest of the sentence is in effect built into the higher-order operator entry for the relative pronoun of type (N\N)↾(S↾NP) in (37). We now consider how this analysis can be extended to pied-piping.

Pied-piping, whimsically named in Ross 1967, 24, is a species of extraction in which a *wh*-pronoun does not directly correspond to a gap within the relative clause but is itself a subconstituent of a larger fronted constituent corresponding to the gap. The following data exemplify the most basic kinds of pied-piping:

(53) a. the guy [to **whom**] John spoke __ yesterday
b. the guy [to **whose** office] John walked __ yesterday
c. the guy [to **whose** sister] John spoke __ yesterday

More elaborate cases can be found, including Ross' example, which makes it clear that the *wh*-word can be embedded arbitrarily deeply.

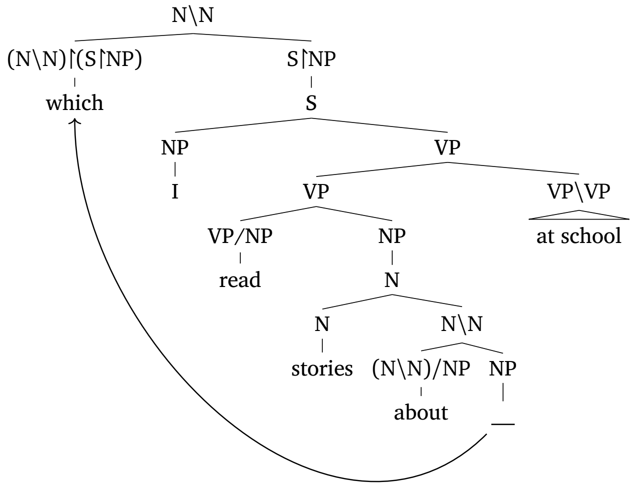
(54) the reports [[the height of the lettering on the covers of **which**] [the government prescribes __]]

Note that the semantic interpretation of pied-piping examples is exactly the same as the corresponding simpler examples in which only the *wh*-word is displaced:

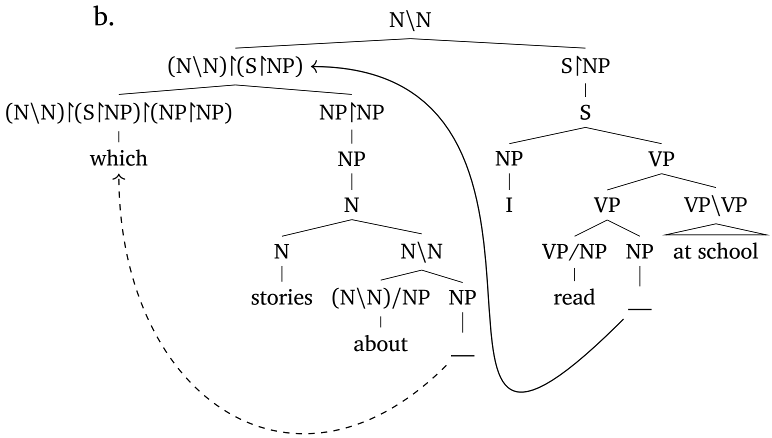
(55) a. Castle Combe is the town [stories about **which**] I read __ at school.
b. Castle Combe is the town **which** I read stories about __ at school.

This correspondence can be graphically represented in the following informal pictures (‘overt’ and ‘covert’ movement is represented by solid and dashed lines respectively):

(56) a.



b.



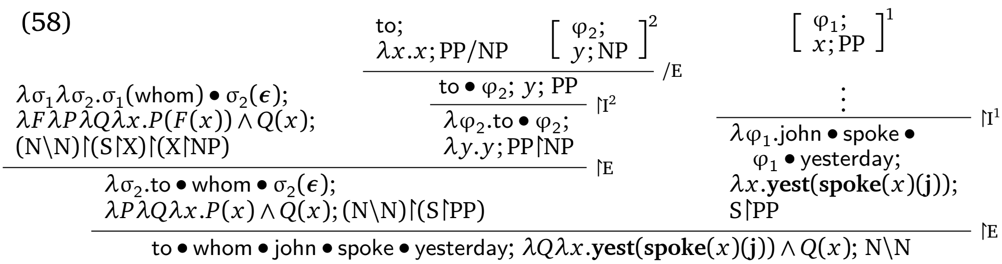
In the case of non-pied-piped relativization (56a), the filler and the gap have the same syntactic type. In contrast, in the pied-piping example (56b), the *wh*-pronoun that triggers relativization is embedded inside the filler, and it is this entire filler phrase that ‘binds’ the gap in the body of the relative clause. Here, as alluded to by the use of different types of ‘movement arrows’, the correspondence between the gap and the filler is a case of ‘overt movement’, just as with non-pied-piped relativization. By contrast, the identification of the whole *wh*-phrase that contains the *wh*-word as the ‘operator’ that triggers relativization is mediated by a ‘covert movement’-like operation. In the latter, the string of the *wh*-word is embedded inside the filler phrase.

This can be formalized precisely by modifying the lexical entry for the *wh*-operator as in (57) (the key idea here is due to Morrill (1994)).

(57) $\lambda\sigma_1\lambda\sigma_2.\sigma_1(\text{whom}) \bullet \sigma_2(\epsilon)$;
$\lambda F \lambda P \lambda Q \lambda x.P(F(x)) \wedge Q(x); (\text{N}\backslash\text{N})\upharpoonright(\text{S}\upharpoonright\text{X})\upharpoonright(\text{X}\upharpoonright\text{NP})$

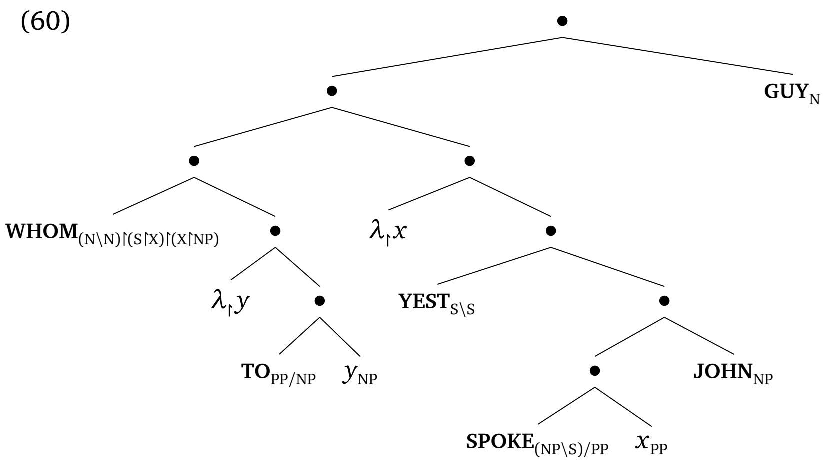
This says that the relative pronoun takes two arguments, some expression of type X missing an NP and an S missing an X, and then becomes a nominal modifier. A sample derivation for (53a) using this entry is shown in (58) (in natural deduction) and (59)/(60) (in the proof term format). Here, since the fronted phrase is a PP, X is instantiated as PP.

(58)



(59) $\textbf{WHOM}_{(\text{N}\backslash\text{N})\upharpoonright(\text{S}\upharpoonright\text{X})\upharpoonright(\text{X}\upharpoonright\text{NP})}$
$(\lambda_{\upharpoonright} y.\textbf{TO}_{\text{PP/NP}}(y_{\text{NP}}))(\lambda_{\upharpoonright} x.\textbf{YEST}_{\text{S}\backslash\text{S}}(\textbf{SPOKE}_{(\text{NP}\backslash\text{S})/\text{PP}}(x_{\text{PP}})(\textbf{JOHN}_{\text{NP}})))$

(60)



Note that this analysis involves two instances of hypothetical reasoning, corresponding to the 'overt' and 'covert' movement operations in the informal diagram in (56b). The hypothetical reasoning with the PP (indexed 1 in (58) and x_{PP} in (59)/(60)) is for forming a gapped sentence of type $\text{S}{\upharpoonright}\text{PP}$ that serves as the body of the relative clause. The hypothetical reasoning involving the NP hypothesis (indexed 2 in (58) and y_{NP} in (59)/(60)) is for identifying the location of the relative pronoun inside the fronted constituent *to whom*. The relativization operator defined in (57) fills in an empty string and the string of the relative pronoun (i.e., the string *whom*) in the positions of the two lambda-bound variables φ_1 and φ_2, reflecting the 'overt' and 'covert' movement statuses of the two hypothetical reasoning steps involved. In Hybrid TLG, 'covert' and 'overt' movement are handled by the same formal mechanism, and the difference between the two merely consists in whether an overt string is substituted for the bound variable position in the prosodic function that is given as an argument to the higher-order operator.

Since the 'in-situ' operator relationship between the relative pronoun and the fronted expression containing it is mediated by $\upharpoonright$, we predict that the *wh*-pronoun can be embedded inside the fronted constituent arbitrarily deeply. Thus, Ross's (1967) example can be accounted for in the same way as the simpler PP pied-piping example in (58) above. We show the derivation in proof term notation:

(61) $\textbf{WHICH}_{(\text{N}\backslash\text{N}){\upharpoonright}(\text{S}{\upharpoonright}\text{X}){\upharpoonright}(\text{X}{\upharpoonright}\text{NP})}$

$$(\lambda_{\upharpoonright} y.\textbf{THE}_{\text{NP/N}}(\textbf{HEIGHT}_{\text{N/PP}}(\textbf{OF}_{\text{PP/NP}}(\textbf{THE}_{\text{NP/N}}(\textbf{ON}_{(\text{N}\backslash\text{N})/\text{NP}}(\textbf{THE}_{\text{NP/N}}(\textbf{COVERS}_{\text{N/PP}}(\textbf{OF}_{\text{PP/NP}}(y_{\text{NP}}))))(\textbf{LETTERING}_{\text{N}}))))))$$

$$(\lambda_{\upharpoonright} x.\textbf{PRESCRIBES}_{(\text{NP}\backslash\text{S})/\text{NP}}(x_{\text{NP}})(\textbf{THE}_{\text{NP/N}}(\textbf{GVT}_{\text{N}})))$$

$$= \mathsf{the} \bullet \mathsf{height} \bullet \mathsf{of} \bullet \mathsf{the} \bullet \mathsf{lettering} \bullet \mathsf{on} \bullet \mathsf{the} \bullet \mathsf{covers} \bullet \mathsf{of} \bullet \mathsf{which} \bullet \mathsf{the} \bullet \mathsf{government} \bullet \mathsf{prescribes};$$
$$\lambda P \lambda x.P(x) \wedge \textbf{prescribe}(\textbf{the}(\textbf{height}(\textbf{the}(\textbf{on}(\textbf{the}(\textbf{covers}(x)))(\textbf{lettering})))))(\textbf{the}(\textbf{gvt})); \text{N}\backslash\text{N}$$

Here, X is instantiated as NP. The question of which syntactic type can be pied-piped is a rather thorny issue. As noted by Arnold and Godard (2021), even a descriptively correct generalization for a well-studied language like English is unclear. We won't attempt to address this issue, since the analysis of pied-piping itself is not our central goal.

5 EXTRACTION PATHWAY MARKING AS PROOF STRUCTURE MARKING

Having reviewed the system of Hybrid TLG, we are now ready to present the full formal analysis of EPM. We start our illustration with the Iringlish case in Section 5.1 (which is mostly a review of the proposal already presented in Section 3.2). This is followed by an illustration of a wider range of options that other languages exploit for the purpose of EPM encoding (Section 5.2). Here, we focus in particular on the floating quantifier *all* in Irish English and information structure-sensitive word-order encoding in Dinka, while touching on various related strategies displayed by other languages along the way. This discussion is meant to demonstrate that our proof-theoretic reconceptualization of the notion of cyclicity has a broad empirical coverage with some interesting semantically-oriented typological implications (discussed briefly in Section 6). Section 5.3 then briefly considers implications for other phenomena pertaining to cyclicity such as reconstruction effects. The final part of this section (Section 5.4) offers a brief comparison with an approach to EPM in HPSG, which dispenses with cyclic movement but encodes the effect by feature propagation. We believe that the discussions in this section will clarify further the ways in which our approach inherits the key ideas of the earlier accounts as well as ways in which it can be seen to offer new insights.

5.1 *Accounting for extraction pathway marking*

Since we have already presented the analysis of Iringlish in informal terms in Section 3, here, for the most part we just reproduce the formal lambda terms corresponding to the informal tree diagrams in Section 3.3. This is followed by some additional discussions of residual issues (on Iringlish and other languages).

The proof term notation for the tree in (20) (for (62)) can be written as (63).

(62) the man **aL** [I said **aL** [I thought **aL** [__ would be there]]]

(63) $\mathbf{REL}_{(N\backslash N)\upharpoonright(S'\upharpoonright NP_{+wh})}$
$(\lambda_{\upharpoonright}x.\mathbf{AL}_{S'/S}(\mathbf{SAID}_{(NP\backslash S)/S'}$
$(\mathbf{AL}_{S'/S}(\mathbf{THOUGHT}_{(NP\backslash S)/S'}$
$(\mathbf{AL}_{S'/S}(\mathbf{WBT}_{NP\backslash S}(x_{NP_{+wh}})))(\mathbf{I}_{NP})))(\mathbf{I}_{NP})))$

Here, each token of *aL* applies to a clausal complement containing a free NP_{+wh} variable, and hence is legal.

The bad cases in (22) can be reproduced in the form of proof terms as in (64).

(64) a. $\mathbf{REL}_{(N\backslash N)\upharpoonright(S'\upharpoonright NP_{+wh})}$
$(\lambda_{\upharpoonright}x.\mathbf{AL}_{S'/S}(\mathbf{SAID}_{(NP\backslash S)/S'}$
$(\mathbf{AL}_{S'/S}(\mathbf{THOUGHT}_{(NP\backslash S)/S'}$
$(\mathbf{GON}_{S'/S}(\mathbf{WBT}_{NP\backslash S}(x_{NP_{+wh}})))(\mathbf{I}_{NP})))(\mathbf{I}_{NP})))$

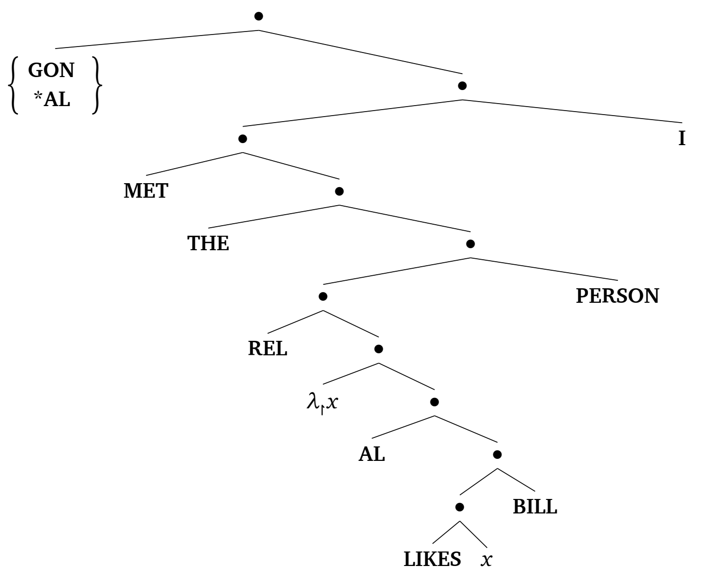
b. $\mathbf{REL}_{(N\backslash N)\upharpoonright(S'\upharpoonright NP_{+wh})}$
$(\lambda_{\upharpoonright}x.\mathbf{GON}_{S'/S}(\mathbf{SAID}_{(NP\backslash S)/S'}$
$(\mathbf{AL}_{S'/S}(\mathbf{THOUGHT}_{(NP\backslash S)/S'}$
$(\mathbf{AL}_{S'/S}(\mathbf{WBT}_{NP\backslash S}(x_{NP_{+wh}})))(\mathbf{I}_{NP})))(\mathbf{I}_{NP})))$

A further prediction of this approach is that when extraction terminates in an embedded clause, the complementizer in a higher structure will be *goN*, rather than *aL*. We illustrate this point with the following (artificial) example:

(65) I said $\left\{\begin{array}{c}\textbf{goN}\\ {}^{*}\textbf{aL}\end{array}\right\}$ I met the person$_i$ **aL** [Bill likes $\_\__i$].

In (66), the variable *x* corresponding to the trace in the embedded relative clause is bound by the lambda operator in the subterm given as an argument to the relativization operator. Thus, the proof term given as an argument to the topmost *aL/goN* contains no free variable. Hence, only *goN* is allowed in the higher clause.

(66)



A case we did not discuss explicitly in Section 3.2 is adjunct extraction. This is completely parallel to extraction of arguments.[12] In an example such as (67), the extracted adjunct semantically modifies the embedded clause. Thus, a hypothetical clausal modifier of type $(\mathrm{S}\backslash\mathrm{S})_{+wh}$ is posited in the lower clause as in (68)/(69).[13]

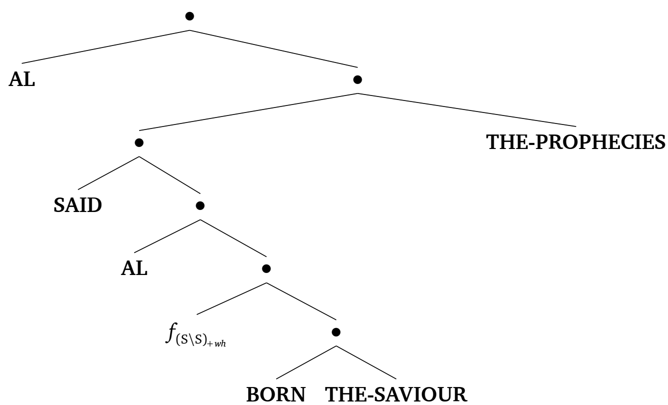
(67) It was in Bethlehem **aL** [the prophecies said **aL** [the Saviour would be born __]]

(68) $\mathbf{AL}_{\mathrm{S}'/\mathrm{S}}(\mathbf{SAID}_{(\mathrm{NP}\backslash\mathrm{S})/\mathrm{S}'}$
$(\mathbf{AL}_{\mathrm{S}'/\mathrm{S}}(f_{(\mathrm{S}\backslash\mathrm{S})_{+wh}}(\mathbf{BORN}_{\mathrm{NP}\backslash\mathrm{S}}(\mathbf{THE\text{-}SAVIOUR}_{\mathrm{NP}}))))(\mathbf{THE\text{-}PROPHECIES}_{\mathrm{NP}}))$

[12] Adjunct extraction poses an interesting theoretical issue in lexicalist theories of syntax such as HPSG and (some variants of) CG (see, e.g., Hukari and Levine 1995), since in such theories, there is an asymmetry between arguments and adjuncts in that the former is an argument of a lexical verb but the latter is standardly a function that takes a verbal projection as an argument. Thus, the pattern in (62) presents a non-trivial issue for a feature-percolation analysis sensitive to valence information of the sort briefly discussed in Section 5.3 below (see Bouma *et al.* 2001 and Levine and Hukari 2006 for details).

[13] Admitting the syntactic type $(\mathrm{S}\backslash\mathrm{S})_{+wh}$ necessitates a move in the underlying theory in which not just atomic types but also complex types can be specified for (at least certain) syntactic features. This may involve some major reworking of the feature system in TLG, but we leave this task for future work.

(69)



Here again, until the variable f (of type $(\mathrm{S}\backslash\mathrm{S})_{+wh}$) is bound, the right form of the complementizer is *aL*, so it is correctly predicted that the two occurrences of *aL* in (67) cannot be replaced by *goN*.

The analysis of complementizer marking in Irish presented above exploits the fact that 'movement' phenomena are analyzed by hypothetical reasoning in TLG and that unwithdrawn hypotheses can be formally treated as unbound variables in the lambda calculus representing proofs. The same approach can be directly extended to cases in which EPM is registered by phenomena that affect the 'clause structure', such as the inversion strategy in Belfast English (and perhaps in French, too, but see the critique of Kayne and Pollock 1978 by Bonami *et al.* (1999)).

(70) What did John say [$_{\mathrm{CP}}$ __ **did** Mary claim [$_{\mathrm{CP}}$ __ **had** John feared __]]?

Assuming that Henry's (1995) characterization of the empirical facts is correct, Belfast English registers extraction pathways by subject-auxiliary inversion consistently.

In lexicalist theories of syntax such as categorial grammar, the standard analysis of inversion involves lexical encoding of the inverted order in the syntactic type of the auxiliary verb (Gazdar *et al.* 1982; Sag *et al.* 2020; Kubota and Levine to appear). For example, in addition to the uninverted, normal word-order variant in (71a) (in which an auxiliary essentially takes a nonfinite VP and returns a finite VP), we have a lexically related alternative entry in (71b) in which it combines with the subject first before combining with its nonfinite VP complement.

(71) a. had; $\lambda F.F$; $(NP\backslash S_{fin})/(NP\backslash S_{bse})$
b. had; $\lambda F.F$; $S_{inv}/(NP\backslash S_{bse})/NP$

The registering of EPM via inversion is straightforward in this type of lexicalist analysis of auxiliaries. In Belfast English, the auxiliary verb entries of the sort in (71) come with additional restrictions that reference the existence of free variables in their $NP\backslash S_{bse}$ syntactic arguments, just like the two complementizer forms in Irish in (18).

Extraction pathway marking in other languages 5.2

Having provided an analysis of the basic patterns of EPM, we now turn to the question of whether this analysis is fully general. For this purpose, we critically examine the recent claim by van Urk and Richards (2015) and van Urk (2020) that *both* successively cyclic movement and feature percolation are needed to capture the entire patterns of EPM. According to van Urk and Richards (2015), the crucial piece of evidence comes from the patterns displayed by Dinka. The apparent violation of the V2 word order in the language exceptionally observed at *wh*-extraction pathways provides evidence for actual movement of the *wh*-phrase. However, the 'long-distance' plural agreement cannot be accounted for by movement alone, and requires a feature checking (or feature percolation) mechanism of some sort. Van Urk (2020) summarizes facts from a wider range of languages for each type of evidence.

To state the conclusion first, while we agree with these authors that these phenomena call for some mechanism in the grammar for keeping track of the identity of the gap before the filler-gap linkage is established, the relevant facts can be analyzed adequately by what we have already proposed, together with independently motivated properties of the specific morpho-syntactic phenomena that exhibit EPM effects. Among the two types of alleged evidence for distinct mechanisms, the 'feature checking' evidence can be dealt with by a slight extension of the analysis of the Irish complementizer marking pattern. We briefly demonstrate this point in Section 5.2.1. After that, we turn to the main task in this section, focusing on two types of 'movement evidence' reported in van Urk and Richards 2015 and van Urk 2020, specifically, Dinka word order (Section 5.2.2) and Irish English floating quantifier *all* (Section 5.2.3). Importantly, a key component of van Urk and Richards's (2015) claim is that Dinka exhibits the 'feature percolation' pattern and the 'movement' pattern within a single language. We counter this claim by showing that the two patterns found in this language (plural marking and word order) can be dealt with by making different lexical items in the language sensitive to essentially the same type of information.

5.2.1 A note on 'agreement' type extraction pathway marking effects

Cases of EPM in which the marking is sensitive to some particular syntactic or semantic feature of the extracted expression, such as the plural marking morphology in Dinka reported in van Urk and Richards 2015, perhaps requires some discussion, before we tackle the main issue of the movement-type evidence for EPM. Here, we show that such cases can be analyzed essentially by the same approach we proposed for Irish complementizer marking, together with the feature-based account of agreement standardly assumed in lexicalist syntax (including TLG).

For the purpose of illustration, suppose that Iringlish had morphological indication of the plurality of the extracted item realized as reduplicative morphology in the form of an intermediate verb. Agreement is handled via features encoded in syntactic categories in lexicalist theories of syntax. Using this feature-based analysis of agreement, a plural-gap variant of the verb *think* can be defined as follows:

(72) $\textsc{thought-pl}_{\mathrm{VP/S'}}$ = thought-thought; **thought**; VP/S′
where for any α, $\textsc{thought-pl}(\alpha)$ is defined only if $fv_{\mathrm{X}_{+wh}}(\alpha) \neq \emptyset$ and the singleton element of $fv_{\mathrm{X}_{+wh}}(\alpha)$ has type NP_{+pl}

(73) the $\left\{ \begin{array}{l} \text{a.*man} \\ \text{b. men} \end{array} \right\}$ aL [I **thought-thought** aL [__ would be there]]

Since the gap NP and the head noun are required to agree in number by the relativization operator, in (73a) the gap NP has type NP_{-pl} and in (73b) it has type NP_{+pl}, yielding the subterms in (74a) and

(74b), respectively, as arguments to (72). Only the former satisfies the definedness condition for (72), correctly capturing the pattern in (73).

(74) a. $\mathbf{AL}_{S'/S}(\mathbf{WBT}_{NP\backslash S}(x_{NP_{-pl}}))$
b. $\mathbf{AL}_{S'/S}(\mathbf{WBT}_{NP\backslash S}(x_{NP_{+pl}}))$

V2 word order in Dinka 5.2.2

Van Urk and Richards (2015) present the following pattern of extraction pathway marking reflected in V2 word order in Dinka as evidence for an actual movement of a copy of the *wh*-phrase in successive cyclicity. We reproduce the relevant pattern in Dinklish, another hypothetical dialect of English which mimicks (the relevant part of) Dinka syntax with an English lexicon.

First, (75) shows that normally embedded clauses exhibit the V2 word order, and that leaving the preverbal position empty is not allowed.

(75) a. Bill$_j$ thinks __$_j$ ke [Mary$_i$ bought __$_i$ the book].
'Bill thinks that Mary bought the book.'
b. *Bill$_j$ thinks __$_j$ ke [__ bought Mary the book].

But there is a systematic exception to this V2 word order requirement. The preverbal position can, and in fact must, be empty when it is crossed by a *wh*-dependency chain. This is demonstrated by (76).

(76) a. Who$_i$ thought John ke [__$_i$ said Mary ke [__$_i$ criticized Bill __$_i$]]?
'Who did John think Mary said Bill criticized __?'
b. *Who$_i$ thought John ke [**Mary**$_j$ said __$_j$ ke [__$_i$ criticized Bill __$_i$]]?
c. *Who$_i$ thought John ke [__$_i$ said Mary ke [**Bill**$_j$ criticized __$_j$ __$_i$]]?
d. *Who$_i$ thought John ke [**Mary**$_k$ said __$_k$ ke [**Bill**$_j$ criticized __$_j$ __$_i$]]?

(76a) is grammatical since the preverbal positions in the most embedded and intermediate clauses are both left unoccupied. By contrast, in the ungrammatical (76b–d), either the preverbal position in the lowest or the intermediate clause (or both) is occupied by an overt NP.

Van Urk and Richards (2015) characterize the preverbal position as Spec,CP. According to them, the pattern in (76) falls out immediately if Spec,CP is an intermediate landing site of the moved *wh*-phrase.

However, there is an alternative account of this distributional pattern that doesn't rely on actual movement of a *wh*-phrase, in which the semantic effect of extraction is taken to be a key component of the explanation. The key idea is that the preverbal position in Dinklish (or Dinka) corresponds to the 'variable' slot in the abstract predicate-argument structure underlying the topic/comment structure in ordinary sentences and the focus/background structure in *wh*-questions. To make this idea more concrete, we make the following assumptions:

(77) a. Every clause must be associated with at most one 'most prominent' element.

b. The preverbal position is the designated position for the prominent element, and is licensed through $\upharpoonright$.

c. As a consequence of (77a,b) when $\upharpoonright$-Introduction applies to produce a predicate-argument structure underlying V2 syntax, there has to be exactly one unwithdrawn hypothesis (corresponding to the element carrying prominence).

To see how this works, consider first the following simple 'Dinklish' sentence with local topicalization:

(78) Bill$_i$ gave __$_i$ Mary the book.

(79)

$$\dfrac{\dfrac{\lambda\varphi_1\lambda\sigma.\varphi_1\bullet\sigma(\epsilon);\ \lambda z\lambda R.R(z);\ S\upharpoonright(S\upharpoonright NP)\upharpoonright NP \qquad \text{bill};\ \mathbf{b};\ NP}{\lambda\sigma.\text{bill}\bullet\sigma(\epsilon);\ \lambda R.R(\mathbf{b});\ S\upharpoonright NP}\upharpoonright E \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{gave};\ \lambda y\lambda x\lambda w.\mathbf{gave}(x)(w)(y);\ S/NP/NP/NP \qquad [\varphi_1;\ v;\ NP]^1}{\text{gave}\bullet\varphi_1;\ \lambda x\lambda w.\mathbf{gave}(x)(w)(v);\ S/NP/NP}/E \qquad \text{mary};\ \mathbf{m};\ NP}{\text{gave}\bullet\varphi_1\bullet\text{mary};\ \lambda x.\mathbf{gave}(\mathbf{m})(w)(v);\ S/NP}/E \qquad \begin{matrix}\vdots\\ \text{the}\bullet\text{book};\ \iota(\mathbf{book});\ NP\end{matrix}}{\text{gave}\bullet\varphi_1\bullet\text{mary}\bullet\text{the}\bullet\text{book};\ \mathbf{gave}(\mathbf{m})(\iota(\mathbf{book}))(v);\ S}/E}{\lambda\varphi_1.\text{gave}\bullet\varphi\bullet\text{mary}\bullet\text{the}\bullet\text{book};\ \lambda v.\mathbf{gave}(\mathbf{m})(\iota(\mathbf{book}))(v);\ S\upharpoonright NP}\upharpoonright I^1}{\text{bill}\bullet\text{gave}\bullet\varphi\bullet\text{mary}\bullet\text{the}\bullet\text{book};\ \mathbf{gave}(\mathbf{m})(\iota(\mathbf{book}))(\mathbf{b});\ S}\upharpoonright E$$

(80) $\mathbf{TOP}_{S\upharpoonright(S\upharpoonright NP)\upharpoonright NP}$
$(\mathbf{BILL}_{NP})(\lambda_{\upharpoonright}x.\mathbf{GAVE}_{S/NP/NP/NP}(x_{NP})(\mathbf{MARY}_{NP})(\mathbf{THE\text{-}BOOK}_{NP}))$

At the step $\upharpoonright$-Introduction applies, there is exactly one free variable x (corresponding to the unwithdrawn hypothesis indexed 1), so the derivation succeeds. Since this hypothesis corresponds to the subject argument of the verb *gave*, we get a subject topicalization sentence.

Consider next the following minimal pair (= (75)), which shows that an embedded topic position cannot remain empty:

(81) a. Bill thinks __ ke [Mary bought __ the book].
'Bill thinks that Mary bought the book.'

b. *Bill thinks __ ke [__ bought Mary the book].
'Bill thinks that Mary bought the book.'

To account for this pattern (and also the *wh*-dependency patterns below), we assume that the complementizer *ke* has the role of ensuring the condition (77a) above, which can be made explicit as in (82).

(82) *Ke* imposes the restriction that there is exactly one free variable in its complement.

As we show immediately below, in the normal topicalization example, after *ke* checks the existence of a free variable, the variable gets bound by $\upharpoonright$-Introduction as usual, and the result is then fed to the topicalization operator; otherwise, that is, when there is a filler corresponding to an embedded gap in a higher clause, *ke* simply passes the free variable upstairs.

For (81), what goes wrong in (81b) is that at the point *ke* combines with the embedded clause, both of the argument positions are occupied by full NPs as in (83a). This violates the condition on *ke* in (82), hence the derivation fails. By contrast, in the case of the topicalization example (81a), the underlined subproof in (83b) satisfies (82), with the free variable x_{NP} which then gets bound by the topicalization operator that licenses the overt NP *Mary* in the clause initial position.

(83) a. $\mathbf{KE}_{S'/S}(\mathbf{BOUGHT}_{S/NP/NP}(\mathbf{THE\text{-}BOOK}_{NP})(\mathbf{MARY}_{NP}))$

b. $\mathbf{TOP}_{S\upharpoonright(S\upharpoonright NP)\upharpoonright NP}$
$(\lambda_{\upharpoonright}x.\mathbf{KE}_{S'/S}\underline{(\mathbf{BOUGHT}_{S/NP/NP}(\mathbf{THE\text{-}BOOK}_{NP})(x_{NP}))})(\mathbf{MARY}_{NP})$

Assuming that the same constraint is operative in more complex sentences involving long-distance extraction of a *wh*-phrase, the pattern in (76) falls out from the assumptions already made. As noted above, all the preverbal positions in intermediate clauses crossed by filler-gap linkage have to be empty:

(84) a. I wonder who$_i$ thought John ke [__$_i$ said Mary ke [__$_i$ criticized Bill __$_i$]].
'I wonder who John thought Mary said Bill criticized __.'

b. *I wonder who$_i$ thought John ke [Mary$_j$ said __$_j$ ke [__$_i$ criticized Bill __$_i$]].
'I wonder who John thought Mary said Bill criticized __.'

We start with the analysis of the grammatical example (84a). Note first that the subproof for the most deeply embedded clause satisfies both (77) and (82), since it contains exactly one hypothesis x_{NP}.

(85) $\mathbf{KE}_{S'/S}\underline{(\mathbf{CRITICIZED}_{S/NP/NP}(x_{NP})(\mathbf{BILL}_{NP})})$

The same process is repeated in the upstairs clause, yielding (86), again satisfying the relevant conditions at the intermediate clause headed by *said*:

(86) $\mathbf{KE}_{S'/S}(\mathbf{SAID}_{S/NP/S'}(\mathbf{KE}_{S'/S}(\mathbf{CRITICIZED}_{S/NP/NP}(x_{NP})(\mathbf{BILL}_{NP})))(\mathbf{MARY}_{NP}))$

Finally, at the matrix level, the hypothesis is withdrawn to yield S↾NP, which is then given as an argument to the *wh*-operator:

(87) $\mathbf{WHO}_{Q\upharpoonright(S\upharpoonright NP)}(\lambda_{\upharpoonright}x.\mathbf{THOUGHT}_{S/S'/NP}(\mathbf{JOHN}_{NP})(\mathbf{KE}_{S'/S}(\mathbf{SAID}_{S/NP/S'}(\mathbf{KE}_{S'/S}(\mathbf{CRITICIZED}_{S/NP/NP}(x_{NP})(\mathbf{BILL}_{NP}))(\mathbf{MARY}_{NP}))))$

Turning now to the ungrammatical (84b), the offending structure is the subproof for the intermediate clause headed by *said*, where the preverbal position is occupied by the local subject *Mary* of that clause, instead of being left empty. As in the above (81a) (with derivation in (83b)), in order to license an overt NP in the topic position, we need to do hypothetical reasoning as in (88). But the underlined part violates the condition on *ke* in (82), since this subproof has two variables x_{NP} (corresponding to the *wh*-filler) and y_{NP} (for the local topic).

(88) $\mathbf{TOP}_{S\upharpoonright(S\upharpoonright NP)\upharpoonright NP}(\lambda_{\upharpoonright}y.\mathbf{KE}_{S'/S}\underline{(\mathbf{SAID}_{S/NP/S'}(\mathbf{KE}_{S'/S}(\mathbf{CRITICIZED}_{S/NP/NP}(x_{NP})(\mathbf{BILL}_{NP})))(y_{NP})})))$

To summarize, the Dinka V2 word order pattern in (76) (in Dinklish) can be explained by an interaction of the topicalization operator and *wh*-extraction. The ungrammatical cases all violate the constraint that there has to be exactly one 'prominent' element in a clause. Since both topicalization and *wh*-extraction exploit hypothetical reasoning at the syntax-semantics interface to identify a particular expression as the 'prominent' element with respect to the respective constructions (where 'prominent' corresponds to focus in *wh*-extraction and topic in topicalization), we predict the same pattern as van Urk and Richards (2015), without treating the preverbal position as a particular type of syntactic projection targeted by cyclic movement.

Linking the interpretation of a variable to discourse prominence may seem like a stipulative association of a syntactic restriction on semantic interpretation with an information-structural property of a dynamic pragmatic background. But increasingly, it is becoming evident that such associations must be recognized, in the interest of empirical generality. For example, this is precisely the kind of condition that Toosarvandani (2016) identifies as the basis for configurational restrictions on the distribution of Gapping in English. In still more recent work, Barros and Frank (2023) have shown that apparently purely syntactic restrictions on the interpetation of multiple sluicing (for which a phase-based analysis was attempted in an earlier work by Grano and Lasnik (2018)) are best understood in terms of discourse prominence status holding between discourse referents in material separated by a clause boundary. Note in particular here that there is a quite suggestive parallel with our proposal for Dinka: in both analyses, there is a prominence relationship established in higher clauses which determines how a variable – corresponding to a bound pronoun in the English data and a reserved preverbal position in Dinka – can be interpreted. We take this sort of dependency relationship to point to a principled basis for the condition in (82).

Irish English *all* 5.2.3

McCloskey (2000) argues that the Ulster subdialect of Irish English allows the extracted operator *what all* to jettison the quantifier-like *all* at various points along a Spec-to-Spec series of local extraction steps, giving tangible evidence that the extracted *wh*-phrase has passed through those steps to arrive at its final landing site. His evidence for this analysis includes the set of data in (89)–(91).

(89) a. What **all** did you get __ for Christmas?
b. Who **all** did you meet __ when you were in Derry?

(90) a. What did you get **all** __ for Christmas?
b. Who did you meet **all** __ when you were in Derry?

(91) a. What **all** did he say (that) he wanted __?
b. What did he say (that) he wanted __ **all**?
c. What did he say **all** (that) he wanted __?

On McCloskey's reasoning, the semantic identity of the floating and non-floating variants of *what/who all* sentences in (89) vs. (90) justifies an analysis in which *what/who all* is 'underlyingly' a unit. On the other hand, as illustrated in (91), the apparently free-floating *all* appears at exactly the points that correspond either to the *wh*-element's site of origin (as in (91b)) or to an intermediate Spec,CP position on the extraction pathway (as in (91c)). McCloskey then takes the distribution of *all* as (at least indirect) evidence for cyclic movement.

In what follows, we sketch an alternative explanation of these facts which essentially takes *all* to be an adverb, building on Sag and Levine (2006), who offer an argument involving the parallel between Irish English *all* and *exactly/precisely* in Standard American English. We refine the connection between the adverbial syntax of *all* and the semantic effect that it imposes on the interpretation of the fronted *wh*-word, an aspect that remains somewhat vague in the Sag/Levine account. We take *all* to be syntactically a VP adverb which imposes a certain semantic restriction on a free variable in its argument. This latter semantic effect is what gives rise to the apparent synonymy between the floating and non-floating variants of *what/who* … *all*. Here again, our account crucially makes reference to the intermediate status of the proof, in such a way that the semantic interpretation of the free variable (unwithdrawn hypothesis) plays a key role.

One piece of evidence for the assumption that stranded *all* is an adverb comes from data such as the following:

(92) ?What did you put in the drawer __ **all** (yesterday)?

On the VP modifier analysis, the position of *all* in (92) is naturally expected. By contrast, on McCloskey's (2000) movement-based analysis, (92) has to be analyzed as first involving a local movement of *what all* to the post-PP position (which is prohibited for overt, non-*wh*-NPs). However, such an analysis seems implausible given the lack of any independent evidence for the supposed movement operation.[14]

---

[14] Further support for the VP adverb analysis of *all* comes from the distributional parallel between the non-remnant adverb *precisely* and the floating *all*. Note first that *precisely* appears to have a very similar distribution as *all*, occuring in both the post-*wh* position and the 'in-situ' position:

(i) a. What **precisely** do you want __?
 b. What do you want __ **precisely**?

However, as McCloskey himself notes, a closer inspection makes it clear that *precisely* cannot plausibly be analyzed as a *wh*-remnant:

(ii) a. *What did he say yesterday **precisely** that he wanted? [on the same reading as (iib)]
 b. What **precisely** did he say yesterday that he wanted?

If *precisely* were a *wh*-remnant on a par with *all*, then (iia) should have a reading equivalent to (iib), with *precisely* being stranded at an intermediate landing site. However, (iia) clearly lacks such a reading.

Yet despite this clear difference in the *wh*-remnant status, *precisely* and *all* share a remarkable similarity in terms of their syntactic distribution as VP-internal adverbs, as shown by the following examples:

(iii) a. *What did he say {**precisely/all**} to {him/his students} that he wanted to buy __?
 b. ?What did he say to {him/his students} {**precisely/all**} that he wanted to buy __?

This distributional parallel between *precisely* and *all* indicates that the pre-complementizer distribution of *all* that McCloskey takes as sufficient evidence for the Spec,CP remnant status of *all* can be accounted for equally naturally by simply assuming that it is syntactically a VP adverb that obeys the same word-order restrictions as an unequivocally non-remnant *precisely*.

The distributional differences between Irish English *all* and Standard American English *exactly/precisely* with respect to the pre-complementizer positioning in (ii) most likely reflects contrasting low-level prosodic conditions on the place-

For the sake of exposition, we start with the analysis of non-floating (93b) and then extend it to the floating *all* in (93a).

(93) a. Who did Frank tell you **all** that they were after __?
b. Who **all** did Frank tell you that they were after __?

For the non-stranded case, we posit the following entry for *all* as a higher-order modifier for a *wh*-operator (mapping a (Q↾(S↾NP)) to another (Q↾(S↾NP))):

(94) $\lambda\rho\lambda\sigma.\rho(\lambda\varphi.\varphi) \bullet \mathsf{all} \bullet \sigma(\epsilon)$; $\lambda\mathscr{F}\lambda P\lambda x_C.\mathscr{F}(P)(x)$; (Q↾(S↾NP))↾(Q↾(S↾NP))
defined only if the domain set C for x is above the contextually relevant standard for high precision

This may look somewhat complex, but all it does is impose a certain restriction on the interpretation of the semantic variable x bound by the *wh*-operator. The semantic restriction imposed on x dictates that it be chosen from a domain set (i.e., contextually determined set of individuals) C which counts as sufficiently 'precise' in the context in question. By applying (94) to the *wh*-question operator *who* in (95), we obtain (96), which then licenses the semantics (97) for (93b).

(95) $\lambda\sigma.\mathsf{who} \bullet \sigma(\epsilon)$; $\lambda P\lambda x.\mathbf{wh}_{\mathbf{person}}(x)(P)$; Q↾(S↾NP)

(96) $\lambda\sigma.\mathsf{who} \bullet \mathsf{all} \bullet \sigma(\epsilon)$; $\lambda P\lambda x_C.\mathbf{wh}_{\mathbf{person}}(x)(P)$; Q↾(S↾NP)
defined only if the domain set C for x is above the contextually relevant standard for high precision

(97) $\lambda x_C.\mathbf{wh}_{\mathbf{person}}(x)(\mathbf{tell}(\mathbf{you})(\mathbf{after}(x)(\mathbf{they}))(\mathbf{frank}))$
defined only if the domain set C for x is above the contextually relevant standard for high precision

The idea here is that by manipulating the domain set in the direction of increasing precision, things that are normally ignored enter into the domain of entities that the question sentence inquires about. For example, suppose that a police officer is interrogating a witness in an investigation of an issue in which a foregn spy John died after having

ment of modifiers of *wh*-words (of different sizes). We therefore assume, following Sag and Levine (2006), that this distributional difference doesn't affect the plausibility of the adverb analysis of Irish English *all*.

lunch with a suspicious person. In this situation, *What **all** did John eat?* is likely a more appropriate (and less ambiguous) question than *What did John eat?*, and it calls for a higher degree of precision and completeness for a proper answer.

Moving on to the floating *all*, we take this *all* to be syntactically a VP adverb which is reordered to the position immediately preceding the complement clause. This can be dealt with by some kind of surface reordering rule governing adverbs (see, e.g., Kubota 2014 for one approach in TLG), and it is motivated by the parallel distribution between *all* and the 'non-*wh*-remnant' adverb *precisely* noted in footnote 14. We can then take the combinatoric structure underlying the matrix VP in (93a) to be something like the following, where x is the free variable corresponding to the embedded gap:

(98) $\textbf{ALL}_{VP/VP}(\textbf{TELL}_{VP/S'/NP}(\textbf{YOU}_{NP})$
$(\textbf{THAT}_{S'/S}(\textbf{WERE}_{VP/VP}(\textbf{AFTER}_{VP/NP}(x_{NP}))(\textbf{THEY}_{NP}))))$

Floating *all* then has the semantics analogous to the non-floating *all* in (94), with the only difference being that in the case of the floating *all*, the semantic variable that it targets is still *unbound* in the term that it takes as its argument as a VP adverb:

(99) $\textbf{ALL}_{VP/VP}$ = all; $\lambda P.P$; VP/VP
where $\textbf{ALL}_{VP/VP}(\alpha)$ is defined only if all elements
$x_C \in fv_{X_{+wh}}(\alpha)$ are such that the domain set C for x is above
the contextually relevant standard for high precision

This imposes exactly the same restriction as the non-floating *all* on the variable x that the question operator ranges over. We thus obtain the same final translation for (93a) as for (93b), namely, (97). Thus, though the exact way in which *all* contributes its meaning in the compositional process is somewhat different in the two cases, we effectively get the same result as McCloskey (2000), preserving the key insight of his analysis that there is a tight semantic connection between the *wh*-phrase and the stranded adverb *all*, but doing away with the undermotivated assumption that the latter forms a syntactic unit with the former in the underlying structure and is a movement remnant in the surface structure.

The analysis of the distribution and interpretation of floating *all* in Irish English sketched above takes the 'stranded' *all* to be an adverb-like operator that targets the denotation of the free variable in the subproof and imposes an additional restriction on its interpretation. Interestingly, at least some of the cases of EPM reported in the literature of the 'remnant movement' type seem to be amenable to a similar treatment. For example, the 'stranding' of quantifier-like elements in Wolof, reported in Torrence 2018 (cited in Davis 2020), consists of a paradigm such as the following:

(100) a. [F-an **f-eeneen**]$_k$ l-a Ayda wax ne l-a-a dem t_k ?
where other COP Ayda say that cop.1sg go
'Where else did Ayda say that I went?'

b. F-an$_k$ l-a-nu foog [t_k **f-eeneen**]$_j$ ne la-a togg-e
where cop.3pl think other that cop.1sg cook
ceeb t_j ?
rice
'Where else do they think that I cooked rice?'

Here, the 'quantifier-like' element *f-eeneen* that exhibits exceptive interpretation (analogous to English *what else*) restricts the interpretation of the 'trace variable' to things that are not identical to some discourse-salient entity.

A somewhat different pattern is found in Polish, in the following paradigm originally reported by Wiland (2010) (again, we reproduce the data from Davis 2020).

(101) Jaki$_k$ (samochód) Paweł kupił swojej żonie t_k
what car Pawel bought his wife
(samochód)?
car
'What car did Pawel buy his wife?'

(102) a. Jaki$_k$ Paweł kupił [$_{\text{VP}}$ [t_k **samochód**]$_j$ swojej
what Pawel bought car his
żonie t_j]?
wife
'What car did Pawel buy his wife?'

b. Jaki$_k$ Paweł [$_{\text{vP}}$ [t_k **samochód**]$_j$ kupił swojej
what Pawel car bought his
żonie t_j]?
wife
'What car did Pawel buy his wife?'

c. Jaki$_k$ myślisz [$_{CP}$ [t_k **samochód**]$_j$ (*że) Paweł
what think.you car that Pawel
kupił swojej żonie t_j]?
bought his wife
'What car do you think that Pawel bought his wife?'

In these examples, it appears as though the head noun of an extracted *wh*-phrase gets stranded at intermediate landing sites, in an apparent violation of the Left Branch Condition. However, these examples are amenable to a different type of analysis, where the apparently 'stranded' element *samochód* 'car' is again a 'trace-targeting' domain restrictor of some sort, restricting the domain set C to $C \cap$ **car**.

What we can see from the above (including Irish English *all*) is that the fact that some element is semantically related to the *wh*-phrase does not necessarily mean that the expression in question has to form a syntactic unit with the *wh*-phrase at some level of syntactic representation. The alternative analyses we have suggested for these so-called 'remnant stranding' EPM cases crucially exploit the key property of our approach that this phenomenon makes reference to the intermediate status of syntactic derivation/meaning computation involving a hypothetically assumed element. It is interesting to see that items that are 'retooled' for EPM in these languages all have essentially the same semantic function of domain restriction for the targeted variable.

A brief note on cyclicity more generally 5.3

Alongside the EPM effects reviewed above, more abstract types of arguments for the notion of cyclicity have been offered in the literature. We review some of these briefly here, with preliminary remarks about their possible implications for our meaning-centered approach. This class of phenomena are potentially important for a comparison between our approach and the standard configurational approaches as they pertain more directly to the architecture of the syntax-semantics interface. In what follows, we discuss in turn (i) arguments involving reflexive binding; (ii) arguments involving the interactions between reconstruction effects in variable binding and Condition C effects and (iii) arguments involving parasitic gap licensing.

First, the binding pattern of the sort exemplified by (103) has sometimes been adduced in the literature in favor of cyclic movement (see, e.g., Barss 2001).

(103) [Which pictures of himself$_{i/j}$] does John$_i$ think t that Bill$_j$ hates t?

The idea here is that the two trace positions in (103) make available reconstruction sites for the fronted *wh*-phrase containing the reflexive, and choosing one or the other satisfies the local c-command requirement with either *John* or *Bill* as the antecedent. But this argument is quite problematic. As den Dikken (2018) notes, the acceptability of (103) on its two readings is compatible with an alternative, non-configurational account for exempt anaphors (of the sort advocated, e.g., by Pollard and Sag (1992) and Reinhart and Reuland (1993)).

A more elaborate type of argument for cyclic movement has been adduced by authors such as Sauerland (1998), involving reconstruction effects in variable binding and Condition C effects. For example, Sauerland notes that the following contrast due to Lebeaux (1992) can be explained by assuming that Condition C applies at LF and that reconstruction of a moved element to an intermediate landing site is possible (for the purpose of variable binding):

(104) a. [Which paper that he$_k$ gave to Mary$_j$]$_i$ did every student$_k$ think t'_i that she$_j$ would like t_i ?

b. *[Which paper that he$_k$ gave to Mary$_j$]$_i$ did she$_j$ think t'_i that every student$_k$ would like t_i ?

In (104b), the fronted *wh*-phrase has to reconstruct to the most deeply embedded trace position t_i for the pronoun it contains to be bound by a c-commanding quantifier. But this incurs a violation of Condition C. By contrast, in (104a), there is an option for the *wh*-phrase to reconstruct to the intermediate trace position t'_i, which simultaneously satisfies the variable binding condition for the pronoun and Condition C for the R-expression *Mary*.

While offering a full analysis is beyond the scope of this paper, we sketch a possible approach within a TLG setup. Jäger (2005, 174–178) proposes an analysis of reconstruction effects in TLG that has two key components: (i) binding of pronouns and reflexives is mediated by hypothetical reasoning, and reflected explicitly in the syntactic types of the binder and the pronoun, adopting Jacobson's (1999) 'pronouns as identity function' approach; (ii) filler-gap linkage transparently preserves the binding relation, by copying the pronoun-containing status of the filler to the 'gap site' via syntactic type encoding of binding (correlating with semantic type).

On this type of approach, the rough form of derivations for the examples in (104) will look like the following:

(105) a. which f (where $f(x)$ is a paper x gave to Mary) $[\lambda f.$ did every student $[\lambda y.$ think $(\lambda x[$**she** would like $x](\boldsymbol{f}(y)))(y)]]$

b. which f (where $f(x)$ is a paper x gave to Mary) $[\lambda f.$ did **she** think [every student $[\lambda y\ [\lambda z\ [y$ would like $z](\boldsymbol{f}(y))]]]]$

The fronted *wh*-expression receives a functional interpretation involving a person-to-paper mapping f, reflecting its pronoun-containing status. Crucially, this functional variable f (which takes the bound variable y as an argument) has to be introduced in the proof in the most deeply embedded clause in (105b) to enforce binding of the individual variable y by the quantifier. By contrast, in (105a), we can wait till the intermediate clause is built to introduce f since the quantifier appears in the intermediate clause (this is parallel to the availability of the intermediate trace position t_i' in Sauerland's LF-based account). This results in a difference in the structural relationship between the pronoun *she* and the functional variable f (the latter of which gets bound by the fronted *wh*-phrase). Assuming that the R-expression-containing status can be copied from the fronted *wh*-phrase to the f variable (via some feature-matching mechanism, for example) and assuming that Condition C is a condition on the form of the logical proof (parallel to Sauerland's treatment of Condition C as an LF condition), the contrast between (105a) and (105b) follows from the fact that *she* 'c-commands' f in (105b) but not in (105a). While this is still preliminary, it should at least be clear that TLG offers an analysis that preserves the core ideas of Sauerland's LF-based account.

Finally, there is another type of evidence involving parasitic gap licensing due to Nissenbaum (2000).

(106) a. Who did you praise __ to the sky [after criticizing __] [in order to surprise __]?

b. Who did you praise __ to the sky [after criticizing __] [in order to surprise **him**]?

c. *Who did you praise __ to the sky [after criticizing **him**] [in order to surprise __]?

Roughly, the idea is that a moved *wh*-phrase licenses a parasitic gap along the way, in a successive cyclic manner. In (106a), the two adjunct clauses are both inside the largest S hosting the fronted filler, and the *wh*-phrase licenses the gaps inside them as it passes through the stacked vPs. (106b) is different from (106a) in that the outer adjunct clause (*in order to surprise him*) adjoins from outside to a structure in which the filler-gap linkage is completely established. (106c) is the problematic case, in which the offending inner adjunct clause (*after criticizing him*) does not host a gap. The absence of a parasitic gap in the inner adjunct clause prevents cyclic movement of the *wh*-phrase which is required to license the parasitic gap in the outer clause. Nissenbaum takes parasitic gaps to be licensed by an empty operator at LF. This entails that the type of interaction between overt *wh*-movement and parasitic gap licensing in (106) necessitates a 'single cycle' architecture (which abandons the standard T-model) in which overt and covert movement operations are interwoven.

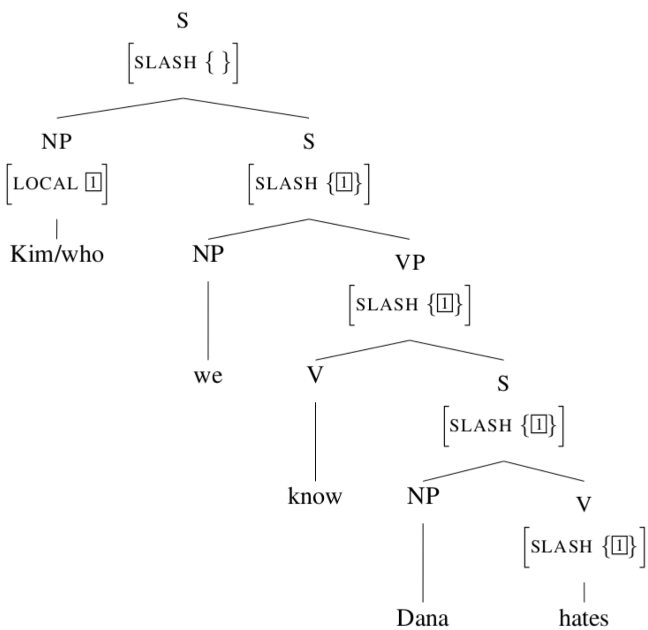
There is an intriguing similarity between the architecture of Hybrid TLG and the 'single cycle' model advocated by Nissenbaum (2000): essentially, Hybrid TLG embodies a 'single cycle' model by design, in that it models overt and covert movement via the same mechanism of prosodic lambda binding within a single model of derivation as logical inference. Interestingly, this architectural design has been independently arrived at without any prior considerations of anything like the Nissenbaum paradigm. This then brings up a question worth exploring in future research: would it be possible to reinterpret the Nissenbaum paradigm within the 'meaning centered' approach we have argued for? Such a reinterpretation would involve viewing both parasitic gap licensing and 'cyclic movement' in processing-oriented terms (the latter along the lines we briefly speculate on at the end of Section 5 on p. 155). We leave this interesting question for future research.

Comparison with a feature-percolation analysis of extraction pathway marking in HPSG 5.4

At this point, the key differences between our proof-theoretic analysis and the successive cyclic analysis standard in derivational approaches should be clear. In the syntactic literature, an alternative to the derivational analysis has been proposed by Bouma *et al.* (2001) in the constraint-based framework of HPSG that makes extensive use of the feature percolation mechanism of the framework. We briefly compare our approach with this HPSG approach in this section.

(107) illustrates the HPSG analysis of extraction.

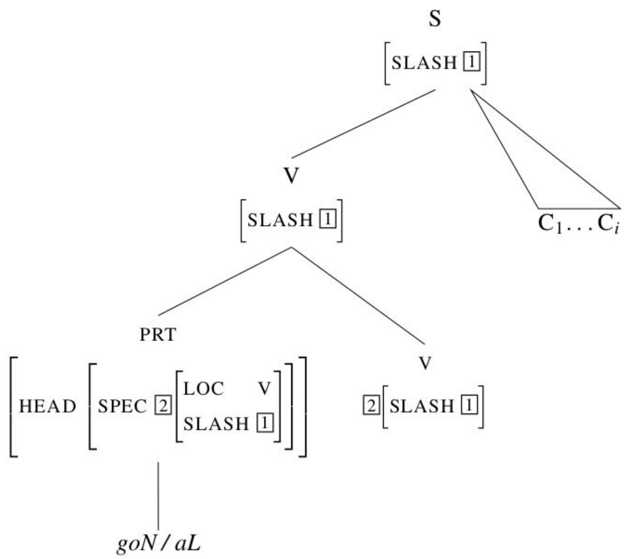
(107)



In HPSG, the SLASH feature is employed for indicating that a phrase contains a gap position (in the object of the verb *hates* in (107)). As in (107), this information is successively inherited from daughter to mother via the feature percolation mechanism inherent to HPSG, until the corresponding filler is found (at the top S node).

Given this general analysis of extraction, in the case of complementizer marking in Irish, the choice of the morphological form of the complementizer can simply be made sensitive to the locally encoded value of the SLASH feature of the verbal projection that *goN/aL* directly combines with, since this feature indicates whether the clause in question contains a gap or not. This is schematically shown in (108).

(108)



The key difference, then, between the TLG analysis and this feature-percolation analysis in HPSG is the following. In the latter, the complementizer choice is dependent on the local syntactic information alone. This is in keeping with the locality condition in HPSG (see, e.g., Sag 2010) and it exploits the general SLASH inheritance mechanism that mediates nonlocal filler-gap linkage via a chain of local feature passing. By contrast, in our TLG analysis, the complementizer choice depends on the existence of an unwithdrawn hypothesis in the subproof (which may be deeply embedded). We have already noted above that this infringes the tenet of direct compositionality in traditional CG, according to which proofs are not representational objects. The reader should now see a connection between HPSG and traditional CG: the CG compositionality thesis roughly corresponds to the locality condition in HPSG – indeed, they are likely to stem from ideas that shaped the common basic form of nonderivational syntactic theories in the 1980s.

While a casual cross-theoretic comparison can be misleading, there does seem to be a tradeoff about which part of the grammar needs to be made complex in the two approaches. Essentially, the HPSG approach abides by the locality principle by slightly enriching the local information encoded at each syntactic node. By contrast, the TLG approach does away with explicit feature percolation at the cost of violating the locality principle in a limited way – limited since all that this approach exploits is a 'filter' constraint that checks the existence of a free variable within a subterm (which conceptually corresponds to the 'tentative assumption' driving hypothetical reasoning in filler-gap linkage).[15] Note that this doesn't involve complex manipulations ('transformations') of the structures of the subterms themselves, or anything that resembles the notion of 'phase' in minimalism (a proof-theoretic analog for this would be a set of meta-constraints imposing an explicit 'control structure' of some sort on proof strategy). In this sense, our proposal is structure-sensitive, but arguably *not* procedural, at least not in the same way that its derivational counterparts (in various avatars of derivational syntax) are.

As a final point of comparison with the constraint-based view of grammar embodied in HPSG, we would like to cautiously bring up possible implications for processing (we ourselves take the competence grammar and the theory of processing to be in principle distinct; see Kubota 2021, Section 5 in this connection). One might initially think that processing-related considerations would favor the local licensing approach embodied in HPSG. However, note that the plausibility of this type of argument largely depends on the assumption that incremental parsing with complex data structures of the sort assumed in HPSG is cognitively realistic. By contrast, TLG embraces a much more indirect relationship between the grammar and processing. That being said, extraction pathway marking formalized as proof structure marking potentially illuminates a possible connection between grammar and processing that has largely been overlooked in the past literature. In proof-theoretic terms, establishing a filler-gap linkage corresponds to withdrawing a hypothesis at a certain point in a proof by finding

[15] In connection to this point, one might recall the discussion from the 'syntax wars' era by proponents of Generative Semantics, e.g., Postal (1972), that global conditions on derivations can always be mimicked by feature marking.

a 'matching' premise (i.e., one that is looking to combine with a conditional statement derived from that hypothesis). Viewing syntactic parsing as proof search – which is a common perspective in TLG – such a complex proof strategy is very likely labor-intensive for the human online parser. It is then not too surprising that some natural languages have developed devices for explicitly flagging the intermediate statuses of the subproofs involved in such proofs, so as to efficiently narrow down the proof search space. Thus, this view offers a particularly natural way of understanding extraction pathway marking as a functionally motivated strategy, one that has fully developed into a grammatically encoded distinction in certain languages. [16]

6 CONCLUSION

We have advocated a new analysis of extraction pathway marking which essentially views this phenomenon as linguistic encoding of proof structure. This has several empirical, technical and conceptual implications that are worth exploring further in future research.

Technically, those familiar with the CG tradition will likely frown on our proposal as it (at least partly) abandons an influential idea of direct compositionality in CG research. We would like to remind such readers that the way our approach makes reference to proof structure is relatively modest, as it merely involves the notion of free variables in a typed lambda calculus (something that is already needed in semantic interpretation anyway). To be sure, global reference to structure is allowed, but we find an analogy to classical Transformational Grammar invoked by one referee somewhat misleading, since, unlike the latter, our approach does not involve arbitrary rewriting of the

[16] One might worry that this discussion on the implications on online processing via proof search might make the relationship between competence and performance obscure and complex in a TLG setting, a point rightly raised by one reviewer. We recognize that this is a legitimate worry, but addressing this important issue fully is a task that we have to leave for future study, in relation to efforts to develop a real processing theory taking some form of TLG as the core component of the competence theory.

structures of already constructed proofs. That being said, we recognize that once this 'Pandora's box' is opened, a question arises as to exactly how much of proof structure reference is allowed and how it is constrained in natural language syntax, an issue we leave for future study. It would also be interesting to see what one can come up with as alternative analyses for EPM within approaches of CG that abide by the notion of direct compositionality more strictly, such as CCG.

Turning to the more conceptual (and empirical) aspects, one might wonder what exactly we gain by this reconceptualization of extraction pathway marking/successive cyclicity. We believe that here the main advantage is that a new, meaning-centered approach to the typology of extraction pathway marking comes into sight, which can be contrasted with the more traditional structure-driven approach that has been dominant in the literature. An almost immediate consequence of our approach is that extraction pathway marking makes reference to the *semantic* relationship between an unwithdrawn hypothesis (corresponding to a free variable) and a larger expression containing it. And there are a couple of 'obvious' choices for encoding such semantic sensitivity in specific morpho-syntactic devices, all attested in one language or another:

- **Direct morpho-syntactic EPM marking** (Irish complementizer selection, Belfast English inversion): This is the most straightforward strategy, in which the language marks the extraction pathway on some functional expression that takes a proposition-denoting constituent as an argument, and signals that the latter involves an incomplete proof.
- **EPM via domain restriction on 'trace' interpretation** (Irish English *all* stranding, Dinka plural marking, Wolof Q-like particle, Polish stranded head N): Impose a restriction pertaining to the semantic interpretation of the relevant free variable. Interestingly, this option seems to allow for more word order freedom than the above morpho-syntactic strategy. This may be due to the fact that domain restrictors are not proposition-taking functions but expressions that are originally part of the (extracted) NP or adverbial elements diachronically.
- **EPM via 'information packaging'** (Dinka V2 word order): This is the most abstract and subtle type of encoding in which the 'distin-

guished' status of the free variable (to be bound by some operator in a higher clause) competes for discourse-oriented prominence. Here again, the semantic interpretation of the variable within the subexpression in which it occurs plays a crucial role in licensing the relevant intermediate proof.

These patterns are of course all well-known, but so far as we are aware, the previous literature does not offer a clear answer to the question of *why* EPM often exhibits sensitivity to the interpretation of the semantic variable with respect to the syntactic context in which it appears. Of course a lot more work needs to be done to investigate this typological literature, but we think that our approach is interesting as it has the potential of shedding a new light on this cross-linguistic typology.

To put the present proposal in a still larger context, it is useful to reflect on the larger goals of comparative syntax in the generative tradition. A core idea behind generative comparative syntax is that the combinatoric system underlying syntax has unique properties characterizing human language. Successive cyclicity has been one major (and quite attractive) candidate for such a property. But a logical reconceptualization of this notion we have attempted in this paper leads to a somewhat different perspective: in our TLG analysis, extraction pathway marking reduces to nothing more than a surface manifestation of an intermediate status of a proof. Our conclusion (and contention), then, is simple: cyclicity may initially look like the best candidate for an unreducible *unique* property of human language, but upon closer inspection, it turns out to be a reflection of a *general* property of logic underlying that system.

ACKNOWLEDGMENTS

We'd like to thank Michael Moortgat, Joanna Nykiel, Shinya Okano and Mizuki Sakamoto for useful discussion. The paper has also benefitted from substantial feedback from the reviewers for *JLM*, and we'd like to thank all of those who were involved in the reviewering process at one stage or another. This work was supported by JSPS KAKENHI Grant Number 21K00541 and the NINJAL collaborative research project 'Toward a Computationally-Informed Theoretical Linguistics', and was carried out as part of a project at the Advanced Language Science (E3P) Research Center at NINJAL.

REFERENCES

Kazimierz AJDUKIEWICZ (1935), Die syntaktische Konnexität, *Studia Philosophica*, 1:1–27, English translation in Storrs MCCALL, editor (1967), *Polish Logic: 1920–1939*, pp. 207–231, Oxford University Press, Oxford.

Doug ARNOLD and Danièle GODARD (2021), Relative clauses in HPSG, in Müller *et al.* (2021), pp. 595–663, `https://langsci-press.org/catalog/book/259`.

Yehoshua BAR-HILLEL (1953), A quasi-arithmetic notation for syntactic descriptions, *Language*, 29:47–58.

Matthew BARROS and Robert FRANK (2023), Attention and locality: On clause-boundedness and its exceptions in multiple sluicing, *Linguistic Inquiry*, 54(4):649–684.

Andrew BARSS (2001), Syntactic reconstruction effects, in Mark BALTIN and Chris COLLINS, editors, *The handbook of contemporary syntactic theory*, pp. 670–696, Blackwell, Oxford.

Cedric BOECKX (2012), *Syntactic islands*, Cambridge University Press, Cambridge.

Olivier BONAMI, Danièle GODARD, and Jean-Marie MARANDIN (1999), Constituency and word order in French subject inversion, in Gosse BOUMA, Erhard HINRICHS, Geert-Jan M. KRUIFT, and Richard OEHRLE, editors, *Constraints and resources in natural language syntax and semantics*, pp. 21–40, CSLI, Stanford.

Robert BORSLEY (2010), An HPSG approach to Welsh unbounded dependencies, in Stefan MÜLLER, editor, *Proceedings of the 17th International Conference on Head-Driven Phrase Structure Grammar*, pp. 80–100, CSLI, Stanford.

Gosse BOUMA, Rob MALOUF, and Ivan A. SAG (2001), Satisfying constraints on extraction and adjunction, *Natural Language and Linguistic Theory*, 19(1):1–65.

Joan BRESNAN and Sam A. MCHOMBO (1995), The lexical integrity principle: evidence from Bantu, *Natural Language and Linguistic Theory*, 13(2):181–254.

Wojciech BUSZKOWSKI (1987), The logic of types, in Jan SRZEDNICKI, editor, *Initiatives in logic*, pp. 180–206, Springer, Dordrecht.

Bob CARPENTER (1997), *Type-logical semantics*, MIT Press, Cambridge, MA.

Rui P. CHAVES and Michael T. PUTNAM (2020), *Unbounded dependency constructions: Theoretical and experimental perspectives*, Oxford University Press, Oxford.

Noam CHOMSKY (1973), Conditions on transformations, in Stephen A. ANDERSON and Paul KIPARSKY, editors, *A Festschrift for Morris Halle*, pp. 232–286, Holt, Rinehart and Winston, New York.

Noam CHOMSKY (1977), On *wh*-movement, in Peter CULICOVER, Thomas WASOW, and Adrian AKMAJIAN, editors, *Formal syntax*, pp. 71–132, Academic Press, New York.

Noam CHOMSKY (1981), *Lectures on government and binding*, Foris, Dordrecht.

Noam CHOMSKY (1986), *Barriers*, MIT Press, Cambridge, MA.

Noam CHOMSKY (2000), Minimalist inquiries: The framework, in Roger MARTIN, David MICHAELS, and Juan URIAGEREKA, editors, *Step by step: Essays on Minimalist Syntax in honor of Howard Lasnik*, pp. 89–156, MIT Press, Cambridge, MA.

Sandra CHUNG (1982), Unbounded dependencies in Chamorro grammar, *Linguistic Inquiry*, 13:39–77.

Barbara CITKO (2014), *Phase theory*, Cambridge University Press, Cambridge.

George N. CLEMENTS, James MCCLOSEY, Joan MALING, and Annie ZAENEN (1983), String-vacuous rule application, *Linguistic Inquiry*, 14:1–17.

Robin COOPER (1983), *Quantification and syntactic theory*, volume 21 of *Synthese Language Library*, Reidel, Dordrecht.

Colin DAVIS (2020), Crossing and stranding at edges: On intermediate stranding and phase theory, *Glossa*, 5(1):17.

Philippe DE GROOTE (2001), Towards abstract categorial grammars, in *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter*, pp. 148–155, Association for Computational Linguistics.

Paul D. DEANE (1992), *Grammar in mind and brain*, de Gruyter, Berlin.

Hans DEN BESTEN (2010), Is there preposition stranding in COMP in Afrikaans? No way!, in Jan-Wouter ZWART and Mark DE VRIES, editors, *Structure preserved: Studies in syntax for Jan Koster*, pp. 57–64, Amsterdam: John Benjamins.

Marcel DEN DIKKEN (2009), Arguments for successive-cyclic movement through SpecCP: A critical review, in *Linguistic Variation Yearbook*, pp. 89–126, John Benjamins.

Marcel DEN DIKKEN (2017), Overtly marked *wh*-paths, in Martin EVERAERT and Henk C. VAN RIEMSDIJK, editors, *The Wiley Blackwell companion to syntax, Second Edition*, John Wiley & Sons.

Marcel DEN DIKKEN (2018), *Dependency and directionality*, Cambridge University Press, Cambridge, UK.

Mark DONOHUE and Anna MACLACHLAN (1999), What agreement in Chamorro?, in Carolyn SMALLWOOD and Catherine KITTO, editors, *Proceedings of AFLA VI*, volume 16, pp. 121–132, University of Toronto Department of Linguistics, Canada: Toronto.

David DOWTY (2007), Compositionality as an empirical problem, in Chris BARKER and Pauline JACOBSON, editors, *Direct compositionality*, pp. 23–101, Oxford University Press, Oxford.

David R. DOWTY, Robert WALL, and Stanley PETERS (1981), *Introduction to Montague semantics*, Reidel Publishing Co., Dordrecht.

Hans DU PLESSIS (1977), *Wh* movement in Afrikaans, *Linguistic Inquiry*, 8:723–726.

Samuel EPSTEIN and T. Daniel SEELY (2002), Rule applications as cycles in a level-free syntax, in Samuel EPSTEIN and T. Daniel SEELY, editors, *Derivation and explanation in the Minimalist Program*, pp. 65–89, Blackwell, Oxford.

Gerald GAZDAR, Geoffrey PULLUM, and Ivan SAG (1982), Auxiliaries and related phenomena in a restrictive theory of grammar, *Language*, 58:591–638.

Thomas GRANO and Howard LASNIK (2018), How to neutralize a finite clause boundary: Phase theory and the grammar of bound pronouns, *Linguistic Inquiry*, 49(3):465–499.

Irene HEIM and Angelika KRATZER (1998), *Semantics in Generative Grammar*, Blackwell, Oxford.

Alison HENRY (1995), *Belfast English and Standard English*, Oxford University Press, Oxford.

Wolfram HINZEN (2012), The philosophical significance of Universal Grammar, *Language Sciences*, 34(5):635–649.

Philip HOFMEISTER and Ivan A. SAG (2010), Cognitive constraints and island effects, *Language*, 86(2):366–415.

William A. HOWARD (1969), The formulae-as-types notion of construction, in Jonathan P. Seldin and J. Roger Hindley (eds.) 1980, *To H. B. Curry: Essays on combinatory logic, lambda calculus, and formalism*, 479–490 (New York: Academic Press).

C.-T. James HUANG (1982), *Logical relations in Chinese and the theory of grammar*, Ph.D. thesis, MIT.

Thomas E. HUKARI and Robert D. LEVINE (1995), Adjunct extraction, *Journal of Linguistics*, 31:195–226.

Pavel IOSAD (2023), Mutation in Celtic, in Peter ACKEMA, Sabrina BENDJABALLAH, Eulàlia BONET, and Antonio FÁBREGAS, editors, *The Wiley Blackwell companion to morphology*, pp. 1565–1606, Wiley-Blackwell, United Kingdom.

Pauline JACOBSON (1999), Towards a variable-free semantics, *Linguistics and Philosophy*, 22(2):117–184.

Gerhard JÄGER (2005), *Anaphora and Type-Logical Grammar*, Springer, Berlin.

Richard KAYNE and Jean-Yves POLLOCK (1978), Stylistic inversion, successive cyclicity and Move NP in French, *Linguistic Inquiry*, 9:595–621.

Robert KLUENDER (1992), Deriving island constraints from principles of predication, in Helen GOODLUCK and Michael ROCHEMONT, editors, *Island constraints: Theory, acquisition, and processing*, pp. 223–258, Kluwer, Dordrecht.

Robert KLUENDER (1998), On the distinction between strong and weak islands: A processing perspective, in Peter CULICOVER and Louise MCNALLY, editors, *The limits of syntax*, volume 29 of *Syntax and Semantics*, pp. 241–279, Academic Press, San Diego.

Yusuke KUBOTA (2014), The logic of complex predicates: A deductive synthesis of ‘argument sharing’ and ‘verb raising’, *Natural Language and Linguistic Theory*, 32(4):1145–1204.

Yusuke KUBOTA (2021), HPSG and categorial grammar, in Müller *et al.* (2021), pp. 1331–1394, `https://langsci-press.org/catalog/book/259`.

Yusuke KUBOTA and Robert LEVINE (2020), *Type-logical syntax*, MIT Press, Cambridge, MA.

Yusuke KUBOTA and Robert LEVINE (to appear), The logic of the English auxiliary system, in *Logical Aspects of Computational Linguistics 2021*, Springer, Heidelberg, `https://ling.auf.net/lingbuzz/006040`.

David LEBEAUX (1992), Relative clauses, licensing, and the nature of the derivation, in Susan ROTHSTEIN and Margaret SPEAS, editors, *Perspectives on phrase structure: Heads and licensing*, pp. 209–239, Academic Press, New York.

Julie Anne LEGATE (1998), Verb phrase types and the notion of a phase, unpublished manuscript, MIT.

Robert D. LEVINE and Thomas E. HUKARI (2006), *The unity of unbounded dependency constructions*, CSLI, Stanford.

Christopher D. MANNING, Ivan A. SAG, and Masayo IIDA (1999), The lexical integrity of Japanese causatives, in Robert LEVINE and Georgia GREEN, editors, *Studies in contemporary phrase structure grammar*, pp. 39–79, Cambridge University Press, Cambridge.

Ora MATUSHANSKY (2005), Going through a phase, in Martha MCGINNIS and Norvin RICHARDS, editors, *MIT Working Papers in Linguistics 49: Perspectives on Phases*, pp. 157–181, MIT Press, Cambridge, MA.

James MCCLOSKEY (1979), *Transformational syntax and model-theoretic semantics*, Dordrecht: Reidel.

James MCCLOSKEY (1990), Resumptive pronouns, Ā binding and levels of representation in Irish, in Randall HENDRICK, editor, *The Syntax of the modern Celtic languages*, pp. 199–248, New York: Academic Press.

James MCCLOSKEY (2000), Quantifier float and *wh*-movement in an Irish English, *Linguistic Inquiry*, 31:57–84.

James MCCLOSKEY (2002), Resumption, successive cyclicity, and the locality of operations, in Samuel EPSTEIN and T. Daniel SEELY, editors, *Derivation and explanation in the Minimalist Program*, pp. 184–226, Wiley.

Vedrana MIHALIČEK and Carl POLLARD (2012), Distinguishing phenogrammar from tectogrammar simplifies the analysis of interrogatives, in Philippe DE GROOTE and Mark-Jan NEDERHOF, editors, *Formal Grammar 2010/2011*, pp. 130–145, Springer, Heidelberg.

Michael MOORTGAT (2011), Categorial type logics, in Johan VAN BENTHEM and Alice TER MEULEN, editors, *Handbook of logic and language*, pp. 95–179, Elsevier, Amsterdam, 2 edition.

Michael MOORTGAT (2014), Typelogical grammar, in Edward N. ZALTA, editor, *The Stanford Encyclopedia of Philosophy (Spring 2014 Ed.)*, Stanford University, available at `https://plato.stanford.edu/archives/spr2014/entries/typelogical-grammar/`.

Glyn MORRILL (1994), *Type Logical Grammar: Categorial logic of signs*, Kluwer, Dordrecht.

Gereon MÜLLER (2010), On deriving CED effects from the PIC, *Linguistic Inquiry*, 41(1):35–82.

Stefan MÜLLER, Anne ABEILLÉ, Robert D. BORSLEY, and Jean-Pierre KOENIG, editors (2021), *Head-Driven Phrase Structure Grammar: The handbook*, Language Science Press, Berlin, `https://langsci-press.org/catalog/book/259`.

Reinhard MUSKENS (2003), Language, lambdas, and logic, in Geert-Jan KRUIJFF and Richard OEHRLE, editors, *Resource sensitivity in binding and anaphora*, pp. 23–54, Kluwer, Dordrecht.

Frederick J. NEWMEYER (2016), Nonsyntactic explanations of island constraints, *Annual Review of Linguistics*, 2:187–210.

Jon NISSENBAUM (2000), *Investigations of covert phrase movement*, Ph.D. thesis, MIT.

Richard T. OEHRLE (1994), Term-labeled categorial type systems, *Linguistics and Philosophy*, 17(6):633–678.

Dennis OTT (2014), Review of Cedric Boeckx, *Syntactic Islands* (CUP, 2012), *Language*, 90(1):287–291.

Carl POLLARD and Ivan A. SAG (1992), Anaphors in English and the scope of binding theory, *Linguistic Inquiry*, 23(2):261–303.

Paul POSTAL (1972), A global constraint on pronominalization, *Linguistic Inquiry*, 3(1):35–59.

Geoffrey K. PULLUM (1992), The origins of the cyclic principle, in J. M. DENTON, G. P. CHAN, and C. CANAKIS, editors, *CLS28: Papers from the 28th Regional Meeting of the Chicago Linguistic Society*, volume 2, pp. 209–236, Chicago Linguistic Society.

Tanya REINHART and Eric REULAND (1993), Reflexivity, *Linguistic Inquiry*, 24(4):657–720.

John Robert ROSS (1967), *Constraints on variables in syntax*, Ph.D. thesis, MIT, reproduced by the Indiana University Linguistics Club.

Ivan SAG, Rui P. CHAVES, Anne ABEILLÉ, Bruno ESTIGARRIBIA, Frank Van EYNDE, Dan FLICKINGER, Paul KAY, Laura MICHAELIS, Stefan MÜLLER, Geoffrey PULLUM, and Thomas WASOW (2020), Lessons from the English auxiliary system, *Journal of Linguistics*, 56:87–155.

Ivan A. SAG (2010), Feature geometry and predictions of locality, in Anna KIBORT and Greville G. CORBETT, editors, *Features: Perspectives on a key notion in linguistics*, pp. 236–271, Oxford University Press, Oxford.

Ivan A. SAG and Robert D. LEVINE (2006), Irish English and the status of intermediate traces, paper presented at the 2006 meetings of the LSA; available at https://www.asc.ohio-state.edu/levine.1/UE.pdf.

Uli SAUERLAND (1998), *The meaning of chains*, Ph.D. thesis, MIT.

Peter SELLS (1984), *Syntax and semantics of resumptive pronouns*, Ph.D. thesis, University of Massachusetts Amherst.

Maziar TOOSARVANDANI (2016), Embedding the antecedent in gapping: Low coordination and the role of parallelism, *Linguistic Inquiry*, 47(2):381–390.

Harold TORRENCE (2018), Left edge agreeing elements in Wolof wh-questions, unpublished manuscript, University of California Los Angeles.

Coppe VAN URK (2020), Successive cyclicity and the syntax of long-distance dependencies, *Annual Review of Linguistics*, 6:111–130.

Coppe VAN URK and Norvin RICHARDS (2015), Two components of long-distance extraction: Successive cyclicity in Dinka, *Linguistic Inquiry*, 46:113–155.

Heinrich WANSING (1992), Formulas-as-types for a hierarchy of sublogics of intuitionistic propositional logic, in David PEARCE and Heinrich WANSING, editors, *Nonclassical logics and information processing*, pp. 125–145, Springer, Berlin and Heidelberg.

Bartosz WILAND (2010), Overt evidence from left-branch extraction in Polish for punctuated paths, *Linguistic Inquiry*, 41:335–347.

Annie ZAENEN (1983), On syntactic binding, *Linguistic Inquiry*, 14(3):469–504.

Yusuke Kubota
0000-0002-8468-5857
kubota@ninjal.ac.jp

NINJAL

Robert Levine
0000-0001-8938-1839
levine.1@osu.edu

Ohio State University

Yusuke Kubota and Robert Levine (2025), Against successive cyclicity: A proof-theoretic account of extraction pathway marking, *Journal of Language Modelling*, 13(1):99–165
https://dx.doi.org/10.15398/jlm.v13i1.339