

The Subregular Complexity of Case and Agreement

A Dissertation Presented

by

Kenneth Jacob Hanson

to

The Graduate School

in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in

Linguistics

Stony Brook University

August 2025

Copyright by
Kenneth Hanson
2025

Stony Brook University

The Graduate School

Kenneth Jacob Hanson

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Thomas Graf

Dissertation Advisor

Associate Professor, Department of Linguistics

Jeffrey Heinz

Chairperson of Defense

Professor, Department of Linguistics

John Bailyn

Professor, Department of Linguistics

Rajesh Bhatt

Professor, Department of Linguistics, University of Massachusetts Amherst

This dissertation is accepted by the Graduate School

Celia Marshik

Dean of the Graduate School

Abstract of the Dissertation

The Subregular Complexity of Case and Agreement

by

Kenneth Jacob Hanson

Doctor of Philosophy

in

Linguistics

Stony Brook University

2025

This dissertation aims to establish a tight bound on the computational complexity of CASE and AGREEMENT dependencies in syntax, that is, the class of possible formal patterns which they exemplify. Such dependencies are shown to be modeled well by a tree-based generalization of of TIER-BASED STRICTLY 2-LOCAL (TSL-2) formal languages, mirroring past results on long-distance phonotactics and syntactic movement.

This characterization has several important consequences. First, TSL-2 is a highly restrictive formal class, ruling out many unnatural and presumably impossible patterns whose absence is left unexplained by earlier, string-based classifications of syntactic patterns. Second, the range of patterns it can express aligns closely with well-known cross-linguistic generalizations and points of variation, unifying many phenomena previously handled by distinct theoretical mechanisms. Third, TSL-2 patterns can be computed by simple, cognitively plausible pattern-detection mechanisms, explaining why it is that language makes use of these and not others. Finally, the class of TSL-2 patterns is in principle efficiently learnable, providing a crucial hint into how human children learn languages so effortlessly, and by extension, how we might design artificial systems that do the same.

All necessary background on subregular formal languages and their application to syntax is included, though some familiarity with Minimalist syntax is assumed. As such, this dissertation may also serve as an introduction to subregular complexity for both theoretical and computational linguists alike.

Contents

List of Figures	vii
List of Tables	viii
List of Abbreviations	ix
Acknowledgments	x
1 Introduction	1
1.1 Why subregular complexity?	2
1.2 Overview of the dissertation	6
2 Subregular Syntax	13
2.1 Formal languages	13
2.2 Computational complexity	17
2.3 Subregular string languages	19
2.4 MG dependency trees	26
2.5 Constraining the derivation	30
2.6 Tree tiers	32
2.7 Tiers over paths	42
2.8 Caveats and limitations	50
2.9 Summary	51
3 Modeling Case	52
3.1 Overview	52
3.2 What is case?	53
3.3 Modes of case assignment	55
3.4 Some theoretical (non-)issues	56
3.5 TSL case assignment – an example	59
3.6 Some technical problems and their solutions	68
3.7 Discussion	71
3.8 Summary	73

4	The Typology of Case	74
4.1	Overview	74
4.2	Spreading	75
4.3	Alignment	77
4.4	Lexical case	80
4.5	Domains and their interactions	81
4.6	Differential argument marking	83
4.7	Non-existent case patterns	85
4.8	The overgeneration question	86
4.9	Summary	87
5	Case in Japanese	88
5.1	Overview	88
5.2	Preview of the analysis	88
5.3	Descriptive generalizations	90
5.4	The analysis	92
5.5	Summary	104
6	Modeling Agreement	105
6.1	Overview	105
6.2	What is agreement?	106
6.3	The complement spine	107
6.4	Constraining agreement	109
6.5	Summary	113
7	The Typology of Agreement	114
7.1	Overview	114
7.2	Invisibility and blocking	115
7.3	Directionality	121
7.4	Multiple probes, one goal	125
7.5	Optionality and multiple goals	129
7.6	Possible and impossible agreement patterns	131
7.7	Summary and discussion	133
8	Agreement in Hindi	135
8.1	Overview	135
8.2	Descriptive generalizations	136
8.3	The two-tier analysis	139
8.4	The typology of agreement, revisited	144
8.5	Summary	146
9	Conclusion	147
	Bibliography	151

List of Figures

1.1	Chapter dependency graph	8
2.1	A tree and its height 2 subtrees	15
2.2	An anchored tree and its subtrees	16
2.3	The Chomsky hierarchy	17
2.4	The subregular hierarchy	18
2.5	Phrase structure tree and dependency tree	28
2.6	Tree and three different tiers	33
2.7	Licit and illicit trees according to a TSL grammar	34
2.8	D-command as a strict partial order	44
2.9	D-command, c-command, and m-command	44
2.10	Command strings as paths	45
3.1	MG dependency tree and case tier	53
4.1	Derivation of nominative displacement by lexical dative	81
4.2	Distribution of ergative case in Basque	83
5.1	Accusative case in Japanese	89
5.2	Examples of licit and illicit verbal case tiers	93
5.3	Example tiers for genitive, lexical dative, and an embedded clause	96
5.4	Verbal case tier for embedded clause with and without ECM	98
5.5	Verbal case tiers for passives of ditransitive	100
5.6	SS-TSL analysis of causative verb	101
5.7	TSL-2 analysis of causative verb	102
6.1	Dependency tree with two spines	108
6.2	TSL analysis of English subject-verb agreement	110
7.1	Case-sensitive agreement in Hindi	116
7.2	Dative intervention in Icelandic transitive expletive construction	119
7.3	Complementizer agreement in Lubukusu	123
7.4	Embedded C and T share a goal in West Flemish	126
7.5	A'-agreement in Dinka	128
7.6	Interaction-satisfaction analysis of optional agreement	130

8.1	Two tiers for Hindi long-distance agreement	139
8.2	Tree, main spine, and agreement tier for long-distance agreement	140

List of Tables

4.1	Summary of case patterns and their TSL-2 analysis	87
7.1	Variants of a TSL-2 grammar and corresponding agreement phenomena . . .	115
7.2	Summary of agreement patterns and their TSL-2 analysis	132
8.1	Parasitic agreement in Hindi-Urdu.	138
8.2	Licit agreement and non-agreement configurations.	142
8.3	Illicit agreement and non-agreement configurations	143

List of Abbreviations

Example glosses are abbreviated as follows:

1	first person	INF	infinitive
2	second person	IPFV	imperfective
3	third person	LOC	locative
ACC	accusative	M	masculine
AUX	auxiliary	N	neuter
C1	class 1	NF	non-finite
C2	class 2	NOM	nominative
CAUS	causative	OV	object voice
CS	construct state	PASS	passive
DAT	dative	PFV	perfective
DEM	demonstrative	PL	plural
ERG	ergative	PRF	perfect
F	feminine	PRT	particle
FIN	finite	PST	past
FUT	future	SG	singular
GEN	genitive	SV	subject voice
HAB	habitual		

Acknowledgments

I would like to thank everyone who helped me, whether directly or indirectly, on the road to completing my Ph.D.

First, thank you to my parents for supporting me in so many ways, big and small, up until now. It is because of you that I made it this far.

Thank you to Cristina Schmitt and Alan Munn, for introducing me to linguistics, to syntax, and to linguistic research; for guiding me towards the completion of a rather ambitious bachelor's thesis; and for making it possible for me to travel to DiGS 2014 in Budapest for my first major conference presentation. Thank you also to Mark McCullen and Joyce Chai for giving me the opportunity to try my hand at teaching and research in computer science.

Thank you to my classmates in the Linguistics B.A. Program at Michigan State University, the rest of the linguistics faculty, and the graduate students for making my first steps into linguistics so memorable. Likewise, thank you to my fellow graduate students in the Linguistics Department at Stony Brook University for being generally awesome. Although the COVID-19 pandemic put a damper on the first year or so, I thoroughly enjoyed spending my time here with you. There are too many of you to name everyone who made a difference, but I hope you will forgive me if I single out Daniel Greeson for being a perpetual lifesaver.

Thank you to the Stony Brook Linguistics faculty for many wonderful classes and conversations, for your endless support and encouragement, and most of all, for believing in me. My thanks especially to Richard Larson, Mark Aronoff, and Tom McFadden for years of great discussions; to Lori Repetti, for being a dependable ally in times of trouble; to Ellen Broselow, for guiding me through my first qualifying paper; to Sandhya Sundaresan, for the seminar that provided the seeds of my analysis of agreement; to Jordan Kodner, for broadening my horizons in computational linguistics; to John Bailyn, for asking the hard questions and helping me formulate my answers; and to Jeff Heinz, for teaching me too many things to list. Thank you also to Rajesh Bhatt for helping me fill in the final piece of this dissertation.

Finally, thank you to Thomas Graf for giving me the tools I needed to succeed, for being patient while I learned how to use them, for pushing me to accomplish far more than I would have thought possible, and for being more or less exactly the Ph.D. advisor I needed.

Chapter 1

Introduction

The world's languages display tremendous variation in all of their dimensions, yet there are also numerous striking regularities. This is true of nearly every aspect of language, and it is certainly true of *CASE* and *AGREEMENT*, which are the topic of this dissertation. Here, I argue that the syntactic side of both phenomena, more specifically, the rules determining which elements in the syntactic structure can enter into case and agreement relations, can in large part be understood in terms of a single underlying computational factor. The hypothesis to be defended is that both dependencies are based in the formal class *TIER-BASED STRICTLY 2-LOCAL*, or *TSL-2*, which has previously been found to be a good model of other long-distance dependencies in syntax as well as phonology.

This result has a number of important consequences, which are outlined below, but one in particular stands out. With respect to language universals and variation, it provides an explanation for the overall locality profile of case and agreement dependencies as well as many specific points of cross-linguistic variation, including selective opacity, case-sensitive agreement, upward agreement, case spreading, differences in case alignment, and several types of differential argument marking. As such, many well-known and seemingly unrelated syntactic phenomena relating to case and agreement, which have previously been handled by unrelated theoretical mechanisms—relativized minimality, probe horizons, dependent case assignment, and more—are reduced to a simple, unified formal model. Not every aspect of case and agreement can be explained in these terms, but those aspects which have a purely formal (i.e. structural) flavor are in large part derivable from this one computational characterization.

The purposes of this chapter are two-fold. First, it provides a high-level introduction to the study of the subregular complexity of syntax, which is relatively young, and in any case unfamiliar to most linguists. Second, it presents an overview of the dissertation, its scope and limitations, and some of the secondary issues that will be addressed in service of the primary claim: case and agreement dependencies in syntax are *TSL-2*. These are the topics of Section 1.1 and 1.2, respectively. As for the remainder of the dissertation, all required formal background is explained starting in Chapter 2, and the core analysis begins in Chapter 3. It is not necessary to read all chapters in order—see Section 1.2.2 for details.

1.1 Why subregular complexity?

This dissertation aims to address, in small part, a very basic question in linguistics: what kind of a thing is language? One perspective (not uncontroversial) treats language as a discrete combinatorial system which assembles structures such as words and sentences in a well-defined manner. But if language is indeed such a system, then *what kind* of combinatorial system is it? Linguists have long noticed that there are striking regularities in the formal patterns occurring within each module of grammar—phonology, morphology, syntax, and semantics—though determining how best to characterize these regularities has proven difficult, with varying approaches having been tried over the decades.

In this work, I take an approach based on COMPUTATIONAL COMPLEXITY, which asks what kind of abstract computations are needed to assemble words and sentences from their atoms. Generally speaking, more complex computational devices are more expressive, which means that they can distinguish a wider range of patterns. The goal, then, is to identify a class of computations (or a group of classes) which is powerful enough to handle the known range of linguistic phenomena, but not so powerful that it can encode patterns that are fundamentally unlike those in any known language. Ideally, this class should not merely define an upper bound on the complexity of linguistic structures; rather, the specific types of structures that it can produce should align closely with known universals and parameters of cross-linguistic variation. Recent research has identified such a class that satisfies both of these criteria: this is the class TIER-BASED STRICTLY LOCAL, or TSL.

1.1.1 String and tree complexity

To fully appreciate the significance of this finding, a bit of context is needed. The analysis of the computational complexity of natural language has a rich history, dating to the pioneering work of Chomsky (1956, 1959). In the traditional approach, linguistic patterns are formalized as sets of strings, called FORMAL LANGUAGES, which are classified into a nested hierarchy according to their complexity (or expressivity). Using this method, it has been determined that syntactic patterns are in general far more complex than those in phonology and morphology. Phonological and morphological patterns are overwhelmingly REGULAR, which among other things means that they can be computed with a finite bound on memory consumption (Kaplan and Kay 1994; Roark and Sproat 2007). In contrast, syntactic patterns are CONTEXT-FREE OR MILDLY CONTEXT-SENSITIVE, requiring significantly more powerful computational resources (Chomsky 1957; Huybregts 1984; Shieber 1985; Joshi 1985).

The idea that computational complexity might differ across modules is not a problem in and of itself, for this could simply indicate that different modules make use of somewhat different underlying cognitive abilities (Heinz and Idsardi 2013). However, the string-based complexity results for syntax also suffer from several limitations. In particular, they do not provide an adequate explanation for the kinds of structural regularities that syntacticians find most compelling. To give just one example, the types of locality restrictions which are characteristic of syntactic dependencies, such as island constraints (Ross 1967) and relativized minimality (Rizzi 1990), do not in any way follow from this classification. Worse still, there

are many types of formal patterns, such as modulo counting (e.g. the number of velar consonants in a word must be a multiple of three), which are within the regular languages yet are essentially unattested across all modules of grammar, including syntax. Thus, this particular complexity analysis might satisfy our first criterion, but not the second.

A solution can be found if we assume the fundamental structures of syntax to be *trees* rather than strings, following common assumptions in contemporary linguistic theory. In this case, syntactic patterns, formalized as sets of trees or TREE LANGUAGES, are also regular (i.e. they can be computed with finite memory), just like those in phonology and morphology (cf. Rogers 1998; Graf 2013).¹ In fact, both phonology/morphology and syntax are SUBREGULAR, as they do not even require the full power of the regular string and tree languages, respectively. This was first observed for phonology, and has been termed the SUBREGULAR HYPOTHESIS by Heinz (2018); also see Heinz (2010a, 2014) and Rogers and Lambert (2019a). Thus, even though phonology and syntax differ in many respects, including their structural representations, the computations performed over these structures may be very similar, or even exactly the same—Graf (2022b) calls this the COGNITIVE PARALLELISM HYPOTHESIS.

But subregularity in and of itself is still a fairly loose upper bound. Many subregular classes have been described in the literature, forming an extensive hierarchy of their own (McNaughton and Papert 1971; Rogers et al. 2013; Lambert 2022, a.o.), and many of these are more powerful than what seems to be necessary for language. Research in SUBREGULAR LINGUISTICS has therefore asked: what subregular class or classes provide the tightest possible classification? Such knowledge is informative since many subregular classes correspond to very simple, cognitively plausible pattern detection mechanisms (Rogers and Pullum 2011; Rogers et al. 2013). Furthermore, the subregular classes that appear most relevant to natural language are precisely those which are efficiently learnable, whether in terms of data and processing requirements (Heinz 2010b, et seq.) or memory burden (Lambert et al. 2021), suggesting an ultimate explanation for some of the basic formal properties of language which is rooted in learnability considerations. (See Section 1.1.3 for further discussion.)

Of the various subregular classes, two stand out as being particularly relevant for language: *local* dependencies (that is, those that are finitely bounded) fall within the class of STRICTLY LOCAL, or SL patterns, while most *long-distance* or *non-local* dependencies (which have no such bound) can be modeled with the superclass TIER-BASED STRICTLY LOCAL, or TSL (Heinz 2018; Graf 2022b).² For example, restrictions on syllable structure are SL over strings, while long-distance phonotactic constraints are TSL; similarly, category selection in syntax is SL over trees, and many long-distance syntactic dependencies, notably movement, are TSL. Even the exceptions which have been identified so far are modeled well by a set of extensions to TSL collectively known as STRUCTURE-SENSITIVE TSL, or SS-TSL (Mayer and Major 2018; Graf and Mayer 2018; De Santo and Graf 2019; Graf and Shafiei 2019).³

¹The specific representation needed to achieve this result is a DERIVATION TREE, which encodes the sequence of steps in the construction of the corresponding PHRASE STRUCTURE TREE. This is explained in Chapter 2.

²The terms *long-distance/non-local* are often used in the syntactic literature to mean *cross-clausal*, but in this work I use them exclusively to mean *unbounded in distance*.

³De Santo and Graf (2019) is the origin of this idea, although it was published later than some of the other works cited.

As will be explained in detail in Chapter 2, SL and TSL provide conceptually simple yet precise notions of linguistic locality. For SL, substrings (or subtrees) of a certain size are categorized as either licit or illicit; a string (or tree) is well-formed iff all of its substructures are licit (Rogers et al. 2013). TSL generalizes the concept of strict locality to long-distance dependencies by defining a TIER of salient elements; all elements not on the tier are ignored, and the rest treated as adjacent (Heinz et al. 2011; Lambert and Rogers 2020). Thus, same basic computations, and by extension, the same notions of locality, apply equally to both phonology and syntax. On top of this, various points of linguistic variation correspond closely to the specific patterns which can be produced using this classes. This is particularly clear for long-distance dependencies, to which we now turn.⁴

1.1.2 The TSL-2 hypothesis

Upon closer inspection, an even more striking parallel emerges. Heinz et al. (2011) show that long-distance phonotactic constraints can be modeled using TIER-BASED STRICTLY 2-LOCAL (TSL-2) string languages, a subclass of TSL in which all constraints reference only two adjacent elements on a tier. McMullin and Hansson (2016) argue that TSL-2 does indeed provide the best fit for the surface constraints resulting from both long-distance harmony and disharmony; McMullin (2016) fleshes out the argument in greater detail. Graf (2022c) argues that a tree-based version of TSL-2 likewise accounts for the types of constraints which characterize syntactic movement, such as islands, while correctly ruling out unnatural constraints.

In addition to providing a tight upper bound on the range of possible patterns, these results also provide sharp insight into the range of attested variation. For instance, the TSL-2 hypothesis explains why we find local harmony, fully unbounded harmony, and long-distance harmony with blocking, but not harmony which is only broken when at least two blocking segments intervene. Similarly, it explains why we find both single and multiple *wh*-movement, why islands exist and why they vary somewhat between languages, and why we do not find movement which can cross one island node but not two. In each case, minor differences in the TSL-2 grammar (that is, the finite description of a TSL-2 formal language) which formalizes the dependency correspond to exactly the kinds of variation we see, while unattested and presumably impossible patterns require greater (or simply different) computational abilities.

In this dissertation, I extend the above arguments to case and agreement dependencies in syntax. I show that TSL-2 model matches their basic locality profile, such that blocking effects such as relativized minimality (mentioned above) and selective opacity (cf. Keine 2019) are unified with both syntactic islands and blockers of long-distance harmony. In addition, I show how the natural parameters of a TSL-2 grammar correspond precisely to well-known phenomena such as dependent case, defective intervention, and case-sensitive agreement, while many other logically simple but linguistically absurd patterns are ruled out on the grounds that they are not TSL-2.

⁴Regarding local dependencies, I argue in Hanson (2024c) that selection, functional hierarchies, and adjunction in syntax can also be understood as SL patterns. Hanson (2025a) fleshes out the details for adjunction.

These findings are significant because there many subregular classes which provide plausible models of at least some long-distance linguistic phenomena. Of these, TSL is tied with SP (strictly piecewise) for being among the simplest (i.e. most restrictive) candidates. Not only do syntax and phonology converge on the same choice, they also fix the constraint window size to the same value of two, the smallest which can relate two elements at all. It would be an extraordinary coincidence if this was not due to some common causal factor, and it is because the clarity and precision of the formal approach to the study of linguistic patterns that we were able to discover this.

That said, some readers may still be wondering: how is this an improvement over conventional linguistic theories? After all, these also aim to balance expressivity and restrictiveness. In the case of long-distance phonotactics, McMullin and Hansson (2016) criticize an Optimality Theory-based framework known as Agreement by Correspondence (Rose and Walker 2004; Hansson 2010) on that the grounds that it easily generates many pathological patterns that are ruled out by the TSL-2 model. Heinz (2018) discusses additional weaknesses of Optimality Theory and related formalisms, focusing on the UR-to-SR map rather than the surface phonotactics. In a nutshell, the mechanism of global optimization which is central to these formalisms makes it extraordinarily difficult to rule out constraint interactions which generate the unwanted patterns. Naturally, the same overgeneration problem applies equally to the use of global optimization in syntax.

And this is just one of several ways in which the TSL-2 model performs well. Another, which I highlighted at the outset of this chapter, is its simplicity and uniformity. For example, the various sub-components of a contemporary version of the Agree operation in Minimalism, defined to respect relativized minimality and parameterized with probe horizons, case-sensitivity, and so on, are essentially unrelated to each other. As I show in Chapter 7, the two ingredients of a TSL-2 computation—relativized adjacency plus a visibility window which contains two elements—subsume all of these and more. Furthermore, these same two ingredients also account for variation in case, movement, and crucially, long-distance phonological dependencies. In contrast, any theory which treats the relevant phenomena as arising from proprietary aspects of syntax (or phonology) is missing a fundamental generalization.

1.1.3 Connections to language cognition and learning

As discussed above, the basic reason why the TSL-2 model is so appealing is that it combines with empirical coverage with formal restrictiveness and theoretical unification. But this is only the beginning, as the same well-understood mathematics behind the model also helps connect linguistic theory and typology to broader concerns of language cognition and learning. Even though these topics are, strictly speaking, beyond the scope of this dissertation, they are worth mentioning as they provide further motivation for adopting this particular computational perspective.

First, as mentioned above, the abstract computations which are needed to recognize and produce TSL patterns are cognitively plausible (Rogers and Pullum 2011; Rogers et al. 2013). Their precise implementation is an open question, as there are several equivalent mathematical characterizations (Lambert and Rogers 2020). Nonetheless, this supports an ultimate

explanation for many formal properties of language based on the types of computations provided by the underlying cognitive system (Graf 2022b). Second, and no less important, is the fact that the class of patterns itself is efficiently learnable; indeed, this could be considered an even more fundamental explanation for the observed typology (Heinz 2010b; Lambert et al. 2021).⁵ This is because even if the human language faculty is capable of producing a certain class of patterns, such patterns will not be reliably transmitted from generation to generation if they cannot be easily learned. Thus, even if there are occasional exceptions to the TSL-2 hypothesis (as mentioned above, this seems to be the case for long-distance harmony), these can potentially be reconciled on learnability grounds, the idea being that the more complex patterns should be susceptible to inter-speaker variation and simplification over time.

Learnability also plays a critical role in addressing the language acquisition problem, namely why it is that children are able to learn languages so quickly and effortlessly (cf. Chomsky 1965). While many details remain to be worked out, there already exist learning algorithms for TSL string languages (Jardine and Heinz 2016; Jardine and McMullin 2017; Lambert 2021), which can in principle be extended to syntax (Hanson 2024d; Swanson 2024). The work that these algorithms do also complements the other components of a full acquisition theory, such as determining when to adopt a proposed generalization (Yang 2016; Heuser et al. 2024). In fact, such algorithms are made possible precisely because of the restrictiveness of the class of patterns to be learned. This fact has been emphasized since the beginning of the subregular program and applies equally to categorical, weighted, and stochastic models alike (cf. Heinz 2009); see Heinz (2016) for a more comprehensive overview.

1.1.4 Summary

To briefly summarize this section, the program of subregular linguistics aims to provide a computational characterization of natural language which takes advantage of recent developments in formal grammar and computational complexity. Compared to previous results based on surface strings, the hypothesis that syntactic patterns are SL or TSL over trees places a much tighter an upper bound on their complexity, and also provides insight into important aspects of linguistic universals and variation, satisfying both of our desired criteria. Going even further, we find that the subclass TSL-2 bounds most long-distance dependencies in both syntax and phonology, suggesting a single causal factor. Looking ahead, the same mathematical tools that brought us this far also open a new route toward an ultimate cognitive explanation for some of the most distinctive properties of language.

1.2 Overview of the dissertation

The primary goal of this dissertation is to extend the TSL-2 analysis of long-distance linguistic dependencies to the syntax of case and agreement. Though simple in conception, there are numerous technical details that must be worked out in order to get the analysis off the

⁵Efficient learnability here refers to learnability in the limit with polynomial time and data (de la Higuera 1997). See Heinz (2016) for an overview of this and other learning paradigms as they relate to linguistics.

ground. Following previous work in subregular syntax, I develop a syntactic framework based on Minimalist Grammars (MGs; Stabler 1997, 2011), allowing the relevant patterns to be formalized in a way that is amenable to a complexity analysis. Additionally, as there are multiple ways in which TSL string languages have been extended to trees, I will explore several approaches. Ultimately, I employ one system for case and another slightly different system for agreement—each chosen to make the analysis as intuitive and linguistically appealing as possible—leaving full unification for future work. In this section, I provide an overview of structure of the dissertation, its empirical scope, and some of the technical issues which are explored in subsequent chapters.

1.2.1 Intended audience

The dissertation is written to accommodate a range of readers, including both theoretical linguists and computational linguists. All necessary background on formal language theory is included, though familiarity with Minimalist syntax and basic discrete mathematics (including sets and functions) is assumed. All mathematical concepts are introduced informally, and less mathematically inclined readers can safely skip the technical details. Similarly, I stick to simple structural analyses and theoretical assumptions of the sort which are commonly accepted in the field of syntax, the configurational approach to case being a possible exception (see Chapter 3). This is primarily for methodological reasons, but has the added benefit that the material should be accessible to those with as little as one rigorous syntax course under their belt and a passing familiarity with modern Minimalist theory, which one might acquire from a standard textbook such as Adger (2003).

1.2.2 Roadmap

The main contents of this dissertation are organized into three parts:

- I. background on subregular syntax
- II. case dependencies and their analysis
- III. agreement dependencies and their analysis

The **background chapter** summarizes the key ideas and past results in subregular phonology and syntax which are relevant to the present studies, serving as a concise introduction to the field. The **next two parts**, focusing on case and agreement, respectively, are each divided into three chapters. **The first chapter** of each part introduces the TSL-2 analysis of the target phenomena, explaining any operational assumptions and touching on some relevant theoretical issues. **The second chapter** is a typological study, showing how the TSL-2 model accounts for well-known phenomena and points of variation while ruling out logically simple yet linguistically unnatural patterns. In doing so, it establishes TSL-2 as an upper bound for the dependencies in question, and also shows how the model explains much of the variation that we see. **The final chapter** of each part is an in-depth analysis of a single language: case in Japanese, and verbal agreement in Hindi. These chapters demonstrate that the model can account for a representative fragment of the syntactic grammars of real languages, not just

isolated patterns without context, and show how it brings clarity and insight into seemingly complex case and agreement paradigms. They are also, to my knowledge, the first such case studies of this kind. I hope that this work will inspire many more such analyses.

A rough dependency graph is shown in Figure 1.1. Each part is largely self-contained, such that readers who are particularly interested in either case or agreement may easily skip the other. Also, while the typological study is ordered before the corresponding case study, the former is not needed to understand the latter, and some readers may prefer to read the case studies first. These weaker dependencies are indicated with dotted lines.

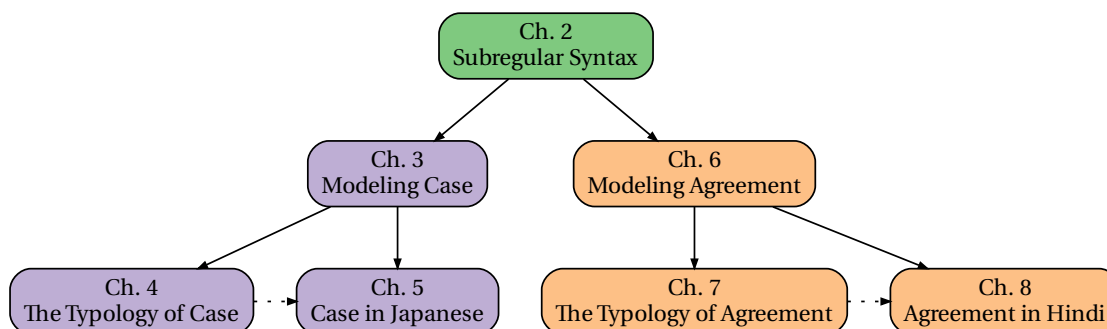


Figure 1.1: Chapter dependency graph

1.2.3 Previous work

Among the works cited in the previous section, a few stand out as providing the direct precedent for the present work. These are:

- McMullin and Hansson (2016) and McMullin (2016), which argue that TSL-2 correctly predicts the locality profile of long-distance harmony and disharmony
- Graf (2022c), which extends the argument to syntactic movement

In addition, Vu et al. (2019) provide a subregular analysis of a range of well-known case patterns, mostly in English. Even though it is not TSL-2 in the strict sense, their analysis has profoundly influenced the present work. In contrast, agreement has received almost no attention in the subregular literature except for a brief sketch by Himmelreich (2022), which I was not aware of when I started this project but which anticipates the general approach.

There is also precedent for modeling of case and agreement in MG more broadly. See Laszakovits (2018) regarding case and Ermolaeva (2018) and Ermolaeva and Kobele (2022) for agreement. I do not build directly on their work here, but discuss it in passing in the relevant chapters.

Finally, much of the present work is based on existing publications and presentations, specifically:

- Chs. 2/6/7 – Hanson (2025c) “Tier-based strict locality and the typology of agreement”
- Chs. 3/4 – Hanson (2025b) “The Typology of TSL Case Assignment”
- Chapter 5 – Hanson (2023) “A TSL Analysis of Japanese Case”
- Chapter 8 – Hanson (2025d) “Two Steps to Parasitic Agreement in Hindi-Urdu”

This is also indicated at the beginning of the relevant chapters. Researchers are encouraged to cite the original papers for the specific claims made therein, and to cite this dissertation for the broader claim that case and agreement are TSL-2.

1.2.4 Why case and agreement?

The present investigation is restricted almost entirely to case and agreement dependencies in syntax, for a variety of reasons. First, previous work in subregular syntax has focused largely on movement (Graf 2018; Graf and Kostyszyn 2021; Graf 2022c, a.o.). Other syntactic dependencies have not been studied nearly as systematically from this perspective, and are in desperate need of it. Next, due to the abstractness of syntactic representations and the degree of cross-linguistic variation, it takes a relatively large amount of work to argue convincingly that a given syntactic dependency is TSL-2. Finally, case and agreement appear to be formally similar to each other and also to movement. This work provides additional evidence that these three dependencies are indeed closely related.

In light of the above, why not focus on just case or agreement, rather than both? This is a fair question, but there are good reasons to address both case and agreement simultaneously. Both phenomena have received significant attention in the theoretical literature in recent decades, resulting in a much clearer picture of the range of patterns which exist as well as theoretical debates about how to best model each phenomenon and whether they might be unified. I believe that the subregular approach has much to say about these issues, as it provides an answer to both why we see so much diversity, and also so much parallelism. Ultimately, this work argues in favor of unification, but at a much wider scale: case and agreement are not only unified with each other, but also movement, not to mention long-distance phonological processes. As such, it is most beneficial to the field to flesh out the TSL-2 analysis of both phenomena to the greatest degree possible within the scope of a single dissertation.

1.2.5 Scope of the investigation

I now explain briefly what I take to fall under the umbrella of CASE and AGREEMENT, respectively. To begin, this work focuses solely on the syntactic side of these phenomena; the details of morphological realization are set aside completely.⁶

Within this scope, case is defined to be a formal category which reflects the context in which nominal phrases (NPs/DPs/etc.) occur in the syntactic structure. These go by names

⁶While morphological processes have been studied from a subregular perspective (Chandlee 2017), the syntax-morphology interface has seen almost no attention. The distinction between syntax and post-syntax is also not completely clear cut; I discuss this issue briefly in Chapter 3.

such as nominative, accusative, genitive, etc, and inform the morphological realization of the head noun and other parts of the nominal phrase. I do not aim to handle licensing of overt nominals—the CASE FILTER of Chomsky 1981. In principle there is no reason why the TSL-2 model could not handle this as well, but it is logically separate, and dependent on the ability to correctly assign/license cases in the first place. Additional details on the conceptualization of case are provided in Chapter 3.

Agreement, in turn, is taken to involve morphological features which covary on elements in the syntactic structure. Here, I focus exclusively on agreement in so-called ϕ -features: person, number, and gender/class. These are pervasive and well-studied; furthermore, I am not aware of any formal difference between ϕ -agreement and agreement of other types. In contrast, I exclude phenomena which do not involve morphological covariation (such as the person licensing condition) or which are plausibly semantic in nature (such as NPI licensing). Similar to the situation with the case filter, these are often unified in Minimalist syntax under the same Agree operation that also underlies agreement proper, but they are conceptually distinct. Further details on the treatment of agreement are discussed in Chapter 6.

My definitions are necessarily very strict due to the bottom-up approach of subregular linguistics, which aims to establish the minimum complexity needed for various linguistic phenomena. To lump together conceptually distinct phenomena before determining whether they have the same complexity would therefore be methodologically backwards. Echoing the immediately preceding section, it is better to provide a proper treatment in future work than to try to cover everything here.

In summary, I am dealing with two syntactic dependencies which directly influence the overt realization of syntactic elements. In addition, movement and NPI-licensing phenomena will be discussed briefly as part of the overview of subregular syntax in Chapter 2, and significant attention will be devoted to interactions between movement, case, and agreement in the main parts of the dissertation.

As a final note, since this work is primarily concerned with the purely formal question of how two or more elements in the syntactic structure can be related to one another, it necessarily leaves aside most issues relating to substance of the elements and features involved, which are also a major concern in recent syntactic theory (Harley and Ritter 2002; Nevins 2007; Pesetsky and Torrego 2007, a.o.). This should not be taken to imply that the TSL-2 model cannot handle such patterns, only that it does not provide any particular insights; indeed, no formal theory can do this. Besides this, there are many other factors which shape syntax in various ways, including aspects of meaning and usage as well as constraints on diachronic development, which likewise require independent explanations. As with the language acquisition problem as discussed above, the TSL-2 model is just one component of a complete theory.

1.2.6 “Agreement” vs “Agree”

It is important to distinguish the claim that the set of structures representing a certain linguistic phenomenon is in some formal class from the claim that a given theory of said phenomena generates structures of said class, though we ultimately want these to align. In

the immediately preceding section, I distinguished several phenomena which we might ultimately want to fold into the theories of case and agreement. In addition to this, case itself commonly thought to be closely linked to agreement, or otherwise triggered by the same Agree operation, as is movement (Chomsky 2000; Pesetsky and Torrego 2001; Adger 2003, a.o.). Furthermore, there is little consensus in the literature regarding the precise details of the Agree operation, as evidenced by the proliferation of extensions and modifications such as multiple Agree (Hiraiwa 2001a; Nevins 2011), cyclic Agree (Béjar and Rezac 2009), failed Agree (Preminger 2014), probe reduction (Béjar and Kahnemuyipour 2017), interaction and satisfaction (Deal 2015), dynamic interaction (Deal 2024), and many others besides these. As such, it should be emphasized that the hypothesis under investigation is “agreement patterns are TSL-2”, not “Agree is TSL-2”. To demonstrate the latter, we would need to formalize a particular version of Agree and determine exactly what kinds of formal patterns it can express. This would be a valuable exercise, as it could help to clarify the differences between competing proposals and might reveal previously unnoticed sources of overgeneration.

1.2.7 Some technical considerations

One important aspect of the formal approach to linguistic analysis is that it forces the analyst to make explicit certain choices about the formal machinery which are often left implicit in conventional approaches. Some of these choices will be inconsequential, such that we simply pick one version and move on, but others matter, sometimes in ways that are orthogonal to the main goal of the investigation. In this work, there are several important formal issues which fall in the latter category. These are addressed in detail in the main body of the dissertation, but deserve to be mentioned upfront.

Multi-tier patterns

TSL formal languages are not in general closed under intersection, which means that a system which contains multiple simultaneous TSL patterns need not be TSL. Instead, there is another class, M[ulti]-TSL, which is defined precisely as the intersection of several TSL languages (De Santo and Graf 2019). So, while I aim to show that individual case and agreement patterns are TSL-2, a more correct way to frame the hypothesis for the entire grammar (at least, the part including movement, case, and agreement) is that it is MTSL-2. This is in some ways a technical detail, for we could have simply defined TSL to cover MTSL from the beginning. But it is also an important distinction, as the typological implications of MTSL are much less well-understood than those of TSL. Another important consideration is that it is not obvious what constitutes an individual case or agreement dependency. At present, the best we can do is to avoid postulating multiple tiers without good motivation, as each tier adds to the degrees of freedom of the analysis.

In practical terms, I have decided to focus my attention on what appear to be simplex dependencies, showing how they can be handled with a single tier, and how the parameters of this model align with the variation that we observe, in line with previous studies of this type. In contrast, I have largely set aside certain phenomena which seem to implicate multiple tiers,

as they require multiple pieces of information to be passed around the structure, each with different locality conditions. These notably include person and animacy hierarchy effects, and are (once again) rich enough to merit a dedicated study. That said, the case studies in Chapters 5 and 8 each make use of multiple tiers, and explore the abilities of an MTSL-2 system to some extent. I also consider interactions between movement, case, and agreement throughout this work.

Two roads to TSL syntax

Another important technical issue is that there are multiple ways in which TSL string languages can be generalized to trees. Past approaches can be grouped into two major categories. One is a direct tree-based analog of TSL string languages (which I refer to as `TREE TIERS`), first introduced by Graf (2018) to model movement and also used by Vu et al. (2019) in their analysis of case. In this work, I continue to use tree tiers to model case, though in a somewhat different manner than past work. In the other approach, introduced by Graf and Shafiei (2019), every node is associated with a path called a `COMMAND STRING`, or `c-string`, and the set of all `c-strings` is constrained using a TSL string language. This system was originally devised to handle NPI licensing and binding, and I adapt it here to model agreement.

These two versions of tree-based TSL are formally distinct and in fact incomparable, meaning that they do not stand in a superclass-subclass relationship. Furthermore, the non-overlapping aspects of each seem to be needed in at least some instances. In the future, we may be able to obtain a unified model (cf. Graf and Hanson 2025). For now, I make use of each as seems most appropriate. Both tree tiers and command strings are introduced in Chapter 2, while the details of the system used for agreement are delayed until Chapter 6.

The syntactic representation

For the syntactic representation, this work makes use of MG dependency trees, which are a particular type of derivation tree. The choice of MGs for the syntactic formalism is mostly a practical one, since MGs are simple and well-understood while providing a reasonable baseline for computation complexity, producing regular derivation tree languages and mildly context-sensitive string languages. In principle, this kind of investigation could also be conducted using another mildly context-sensitive formalism such as TAG or CCG. The use of derivation trees rather than phrase structure trees is critical, since regular phrase structure tree languages cannot generate mildly context-sensitive string languages (at least, not in the usual manner of taking the string yield). Finally, the particular format of dependency trees is also important, though to a lesser extent, as the extra nodes in traditional derivation trees must be filtered out, increasing computational complexity to SS-TSL (Graf 2018). Leaving aside this complication, it seems likely that similar results could be achieved in such a system, and perhaps others not yet considered.

Chapter 2

Subregular Syntax

This chapter is an introduction to the subregular analysis of syntactic patterns. By the end of this chapter, the reader will have an understanding of current research in subregular syntax, which forms the foundation of the studies presented in subsequent chapters.

I begin with an overview of some concepts and findings from formal language and computational complexity theory, motivating the present focus on subregular formal languages (Sections 2.1 and 2.2). I also introduce some classes of subregular string languages which are particularly relevant to natural language, namely SL, SP, and TSL, within some examples from phonology, followed by a discussion of MTSL languages (Section 2.3). Next, I introduce the Minimalist Grammar dependency tree model, used in most previous work on subregular syntax (Section 2.4). This is followed by a short interlude which explains at a high level how the constraints of standard Minimalist Grammars can be understood in terms of subregular tree languages (Section 2.5).

From there, I introduce the two ways in which TSL string languages have been generalized to trees: tree tiers (Section 2.6) and tiers over paths (Section 2.7). Examples are provided for a variety of syntactic constraints, mostly relating to movement, in order to demonstrate how the TSL model accounts for both cross-linguistic variation and its limits. This sets the stage for the analysis of case beginning in Chapter 3; the analysis of agreement incorporates some additional elements, which are introduced in Chapter 6. The chapter closes with a brief discussion of some of the limitations of the current approach (Section 2.8).

2.1 Formal languages

Note: Readers already familiar with formal language theory can skip this section.

In formal grammar, it is common to model linguistic objects such as words or sentences as finite discrete structures, usually strings or trees, but also multidominance trees, autosegmental representations, and others. Linguistic patterns, which distinguish certain words or sentences from others, are modeled as FORMAL LANGUAGES, which are (possibly infinite) sets

Parts of this chapter are based on material from Hanson (2025c) “Tier-based strict locality and the typology of agreement”, particularly Sections 2.2, 2.3, and 2.4. The TSL-2 grammar for NPI-licensing with c-strings and the formalization of relativized minimality in Section 2.7 are new.

of such structures. A `STRING LANGUAGE` is a set of strings, a `TREE LANGUAGE` is a set of trees, and so on. I use the term `LANGUAGE` without further qualification to refer to either a string or tree language when clear from context.

`FORMAL LANGUAGE THEORY` is the field of mathematics which studies such sets. In particular, it studies how such sets can be classified according to their expressivity (the range of patterns that they can encode), and the conditions under which different classes of formal languages can be recognized, that is, how difficult it is to determine whether a particular object is a member of a particular language.

In the remainder of this section, I introduce some key ideas relating to the two kinds of structures used in the present work: strings and trees.

2.1.1 Strings, string languages

A `STRING` is a linear sequence of symbols drawn from an `ALPHABET`, denoted Σ , which is the set of all usable symbols. For our purposes, all strings are finite, as is the alphabet. For example, if our alphabet is $\{a, b, c\}$, then the possible strings over this alphabet include $\{a, aaa, abcabc, cccbbbaaabbccccc\}$ among an infinite number of others. The length of a string is the number of symbols it contains; the lengths of the strings just mentioned are 1, 3, 6, and 15, respectively. We also have an `EMPTY STRING`, denoted ϵ , which has length zero.

In general, a single element of a string corresponds to a segment of a word, or a word in a sentence. So, the word ‘apampa’ consists of six elements drawn from an alphabet which contains at least $\{a, p, m\}$, and the sentence *The cat chased the rat* consists of five elements whose labels are members of $\{\text{the, cat, rat, chased}\}$. In both cases, the symbols of the alphabet are atomic, and have no further structure as far as the mathematics is concerned. Granted, in linguistic theory such objects are usually *not* atomic, but in formal language theory it is conventional to abstract away from such details, as they generally have no effect on computational complexity (an important caveat is discussed in Section 2.8).

Sidebar 2.1: When features matter

Linguists traditionally decompose phonetic segments into features. For example, $[i]$ might be $[+high, -back, -round]$, and $[b]$ might be $[bilabial, +voice, -continuant]$. Similarly, syntactic elements (words/morphemes/heads) also bear features such as major category, tense, and person/number/gender. Although the choice of atomic vs. feature-based representations does not matter much for computational complexity, wider concerns might nonetheless push us towards decomposition. For example, it is generally thought that linguistic constraints/processes target `NATURAL CLASSES`, which are sets of elements that share one or more feature values. Later, we will enforce this to some extent in our TSL grammars, even though the formalism itself does not make this restriction. In addition, featural representations can make a big difference for learning, allowing the target grammar to be learned with far less input data; see Markowska and Heinz (2023, 2025) for an example from phonotactic learning.

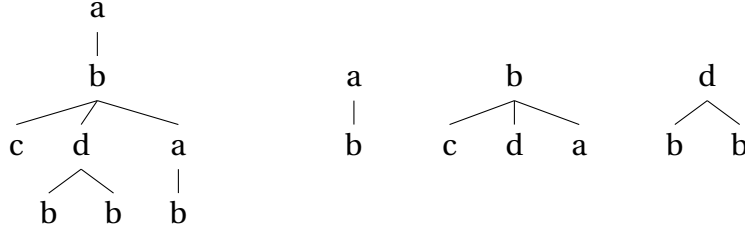


Figure 2.1: A width 3 tree (left) and its height 2 subtrees (right). Some subtrees have a smaller width than the maximum.

In a string, each element has a single **SUCCESSOR**, which is the element that immediately follows it. By taking a contiguous sequence of one or more elements related by the successor relation, we obtain a **SUBSTRING**. For example, the string *abcabc* contains five substrings of length two: *ab*, *bc*, *ca*, *ab*, and *bc*. Of these, three are distinct: $\{ab, bc, ca\}$.

In order to easily refer to the word edge, we augment the representation with the **EDGE MARKERS** $\bowtie$ and $\bowtie$, creating an **ANCHORED WORD MODEL** (the version without the edge markers is an **UNANCHORED MODEL**). Continuing the previous example, the length 2 substrings of newly anchored $\bowtie abcabc \bowtie$ are now $\{\bowtie a, ab, bc, ca, c \bowtie\}$. In this work, I assume all word models to be anchored (even if not notated explicitly), as this simplifies various definitions without affecting computational complexity in a meaningful way.

Substrings are the basis for the definition of several subregular classes of string languages. Of these, the definition for a **STRICTLY LOCAL (SL)** string language is particularly straightforward. Informally, an SL language distinguishes strings according to their substrings of some preset length k , whereby a string is well-formed if and only if all of its length k substrings are well-formed. For example, if we take the earlier set $\{\bowtie a, ab, bc, ca, c \bowtie\}$ as our well-formed substrings, the resulting language includes *abc*, *abcabc*, *abcabcabc* and so on. With slight alternations of the set of licit substrings, we can alter the language in various ways. For example, we could add $a \bowtie$ and remove $c \bowtie$, allowing strings to terminate on an *a* but not *c*, or add *bb*, allowing unbounded sequences of *b*, as in *abcabbbbbbbcbabc*.

Many other linguistically interesting subregular classes are defined in a similar manner, including all of those utilized in this work. In each case, we pick some kind of finite **FACTOR**, or substructure of interest, and specify which factors are licit/illicit. See Section 2.3 for details.

2.1.2 Trees, tree languages

If a string is a structure in which each element has a single successor, then a tree is a generalization of a string in which each element can have multiple successors (cf. Backofen et al. 1995; Rogers 1998). We call this expanded relation the **IMMEDIATE DOMINANCE** relation. For example, the tree on the left of Figure 2.1 has a node labeled *b* which immediately dominates three nodes labeled *c*, *d*, and *a*. It **(PROPERLY) DOMINATES** a total of six nodes, including another three labeled *b*.

In this work, I specifically use **ORDERED TREES**, which means that there is an inherent ordering among the children of each node, call this the **IMMEDIATE LEFT SIBLING** relation. In

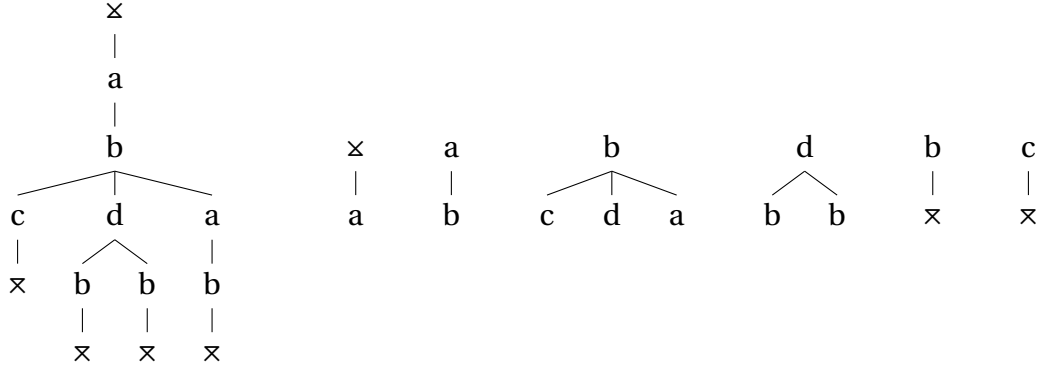


Figure 2.2: An anchored tree (left) and its subtrees (right).

the current example, there is a node labeled c which is the immediate left sibling of a node labeled d , which in turn is the immediate left sibling of one labeled a . While in modern syntactic theory phrase structure trees are commonly held to be UNORDERED, in the dependency representation to be introduced in Section 2.4, sister order indicates c-command (a structural relation), not linear order in the surface string. Thus, there is no real conflict.

Next, we define a SUBTREE of a tree T with height k by taking a node n in T and all of its children, and their children, etc., up to $k - 1$ nodes below n , such that if a descendant of n is included, so are all of its siblings. As an example, all of the unique height-2 subtrees of the tree in Figure 2.1 are listed on the right side. Note that this is different from another common definition of a subtree, which includes *all* of the descendants of a node.¹

A TREE LANGUAGE, as mentioned above, is simply a set of trees. We can therefore define a STRICTLY LOCAL TREE LANGUAGE in a way that is directly analogous to a strictly local string language: there is a set of licit subtrees of fixed height k , such that every tree which is composed only of these subtrees is a member of the language. Note that we usually assume a finite maximum WIDTH, or branching factor, for each tree language. This is already guaranteed for the above definition of a strictly local tree language, since there is necessarily a largest width among the finite set of permitted subtrees, which in turn sets the maximum width for the entire language. Later, in Section 2.6, we will also make use of variable width trees in the definition of a tree tier, though the trees of the language itself still have a maximum width.

As before, we will also augment our tree models with edge markers. In this case, there is a single top edge marker $\times$ at the root of the tree, and a bottom edge marker $\times$ as the child of each leaf. This allows us to directly reference the top and bottom of the tree, just as we can directly reference the left and right edges of a string. The previous example of a tree and its height-2 subtrees is repeated in Figure 2.2. We could also add left and right edge markers to our trees, but this serves no purpose at the moment due to the finite branching factor. Once again, they will become relevant when we get to TSL tree grammars.

¹The present definition is analogous to a substring, while the alternative definition is based on the inductive (recursive) definition of a tree, and is analogous to a string prefix/suffix.

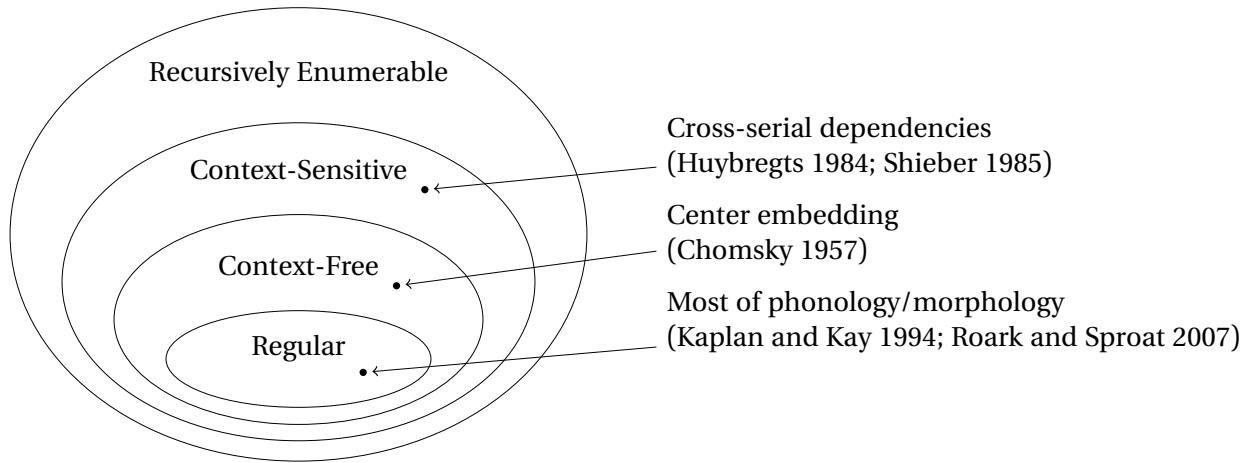


Figure 2.3: The original Chomsky hierarchy (Chomsky 1959), showing inclusion relations among the major classes of string languages.

2.2 Computational complexity

In Chapter 1, I referenced several classes of formal languages which have played an important role in the classification of linguistic patterns. In this section, I describe the lay of the land in a bit more detail.

First, the classic Chomsky Hierarchy (Chomsky 1959), shown in Figure 2.3, classifies string languages according to their expressivity, that is, the range of patterns that they can encode. Each of the major classes corresponds to the set of languages which can be generated by a particular type of STRING REWRITING GRAMMAR. For example, the CONTEXT-FREE LANGUAGES (CFL) are generated by context-free phrase structure grammars of the kind commonly used in introductory syntax classes, in which each rule contains only a single symbol on the left hand side. Each class also corresponds to the set of languages which can be recognized by a particular type of abstract machine. The REGULAR LANGUAGES (REG) are exactly those which can be recognized by a finite state automaton, and the RECURSIVELY ENUMERABLE LANGUAGES (RE) are those which can be recognized by a Turing machine.

When linguistic patterns are classified in this way, syntactic patterns appear far more complex than those in phonology and morphology. The former require some of the capabilities of the CONTEXT-SENSITIVE LANGUAGES (CSL) (Chomsky 1957; Huybregts 1984; Shieber 1985; Kobele 2006), while the latter are almost exclusively regular (Kaplan and Kay 1994; Roark and Sproat 2007), the primary exception being total reduplication, which is also context-sensitive. This basic classification has not changed much in recent years, though many of the finer details have become more clear. A somewhat extended version of the Chomsky Hierarchy is shown in (1) below.

(1) Extended Chomsky Hierarchy

$$\underbrace{\text{FIN} < \dots < \text{SF}}_{\text{"subregular"}} < \text{REG} < \dots < \text{CFL} < \underbrace{\text{TAL} < \text{MCFL} < \text{PMCFL}}_{\text{"mildly context-sensitive"}} < \text{CSL} < \dots < \text{RE}$$

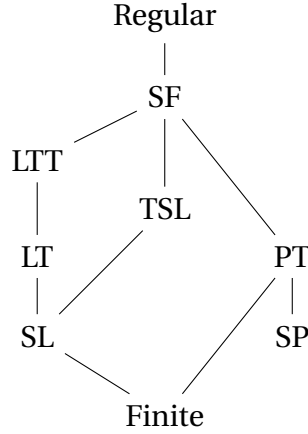


Figure 2.4: The basic Subregular Hierarchy, adapted from Heinz (2018)

Here, several additional classes of linguistic interest are shown, further subdividing the regular and context-sensitive languages. (See Graf 2022b, 2024a for an accessible overview.) First, it has long been thought that syntax does not require the full power of the context-sensitive languages, but rather some subclass of MILDLY CONTEXT-SENSITIVE languages (Joshi 1985), though there are multiple candidates for exactly what class this is. The TREE ADJOINING LANGUAGES (TAL) can encode cross-serial dependencies like in the classic data from Dutch (Huybregts 1984) and Swiss German (Shieber 1985). At present, the PARALLEL MULTIPLE CONTEXT-FREE languages (PMCFL) seem to be an upper bound for the complexity of syntax (cf. Graf 2024a). At the other end of the spectrum, it has become increasingly clear that, to the extent phonology and morphology are regular, they are SUBREGULAR, requiring only very weak subclasses of the regular languages such as SL and TSL (Heinz 2018). Even total reduplication is regular (i.e. finite state) when viewed as a map (Dolatian and Heinz 2018), and in fact subregular (Dolatian and Heinz 2020).

This would seem to reinforce the dichotomy between phonology and morphology, on the one hand, and syntax, on the other. However, many of the types of regular patterns that are not attested in phonology and morphology do not occur in syntax either. Consider modulo counting, which is within the ability of the regular languages. This includes patterns such as the extremely implausible phonological constraint “the number of vowels in a word must be a multiple of three” and the similarly implausible syntactic constraint “the number of prepositions in a sentence must be a multiple of three”. Both of these constraints seem equally unnatural in their respective domains. This suggests that the above view, for all of its insights, is not the full picture.

This brings us to the subregular hierarchy, which further subdivides the regular languages. Of the various subregular classes, there are a handful which have received particular attention in the computational phonology literature, which are shown in Figure 2.4. A proper overview of some of these classes, namely SL, SP, and TSL, is provided in Section 2.3. Also, it should be emphasized that this is just a fraction of the subregular classes that have been described to date; see De Santo and Graf (2019), Rogers and Lambert (2019b), and Lambert (2022, 2023) for various expanded versions of the subregular hierarchy.

For now, consider again the modulo counting examples from above. Such patterns are properly regular, that is, they are beyond the abilities of any subregular class. If phonology is restricted to TSL, then it is no surprise that such patterns are not found. However, assuming that the machinery underlying syntax can handle context-free and even some context-sensitive languages, it is unexpected that such simple constraints seem not just odd, but preposterous. We can construct similar examples of non-existent pairs of patterns which exemplify even subregular classes such as LT (locally testable), for example, “a word must not include both a nasal–stop cluster and vowel hiatus” and its syntactic analog “a sentence may not contain both a verb–pronoun and adverb–adjective pair”. Given that LT and TSL are incomparable in expressivity, this further reinforces the idea that overall computational complexity is not the crucial factor. Rather, *it is as if human language is only sensitive to certain types of patterns, and not others*.

This paradox can be resolved if instead of surface strings, we instead analyze the computational complexity of syntactic patterns in terms of their underlying *structural representations*, which have long been argued to be tree-like. Graf and Heinz (2015) proposed that syntactic patterns would then have the same complexity as other linguistic patterns: they are also TSL, but over trees rather than strings. This has been termed the COGNITIVE PARALLELISM HYPOTHESIS by Graf (2022b).

Even though the focus of this work is on syntax, the string case is easier to understand, and the key mathematical ideas carry over to the tree-based versions. In fact, our tree grammars will be expressed largely in terms of particular strings extracted from the syntactic representation. So, we briefly tour a select portion of the subregular string languages.

2.3 Subregular string languages

The classes of subregular languages which are of interest to us each have several equivalent characterizations. Here, I introduce characterizations based on k -FACTORS, which are finite substructures of a string such as substrings, subsequences, or the most interesting for us, substrings on a tier. By banning strings containing such substructures, we obtain languages in the classes SL, SP, and TSL, respectively. This section provides definitions and examples for each of these, then revisits some aspects of the wider subregular hierarchy.

2.3.1 Notation and terminology

Following standard notation from formal language theory, S^n and S^* denote all strings over set S of length n and of any finite length, respectively. Also, s^n denotes the string consisting of n repetitions of s .

A STRING LANGUAGE is set of strings, possibly infinite. A string language is defined with respect to fixed finite set of symbols called the ALPHABET, denoted Σ . Every string language is therefore a subset of $\mathcal{P}(\Sigma^*)$.

A SUBSEQUENCE of a string s is a string that can be obtained by removing zero or more elements from s without changing their order, and a SUBSTRING is a subsequence of elements which are adjacent in the original string.

Finally, $\bowtie$ and $\bowtie$ are the left and right EDGE MARKERS, which are added to a string so that the beginning, middle, and end can be modeled uniformly.

2.3.2 Strictly local languages and k -factors

An SL grammar is essentially a finite set of FORBIDDEN SUBSTRINGS of some fixed length, and the corresponding SL language is the set of all strings over the alphabet that do not contain any forbidden substrings (McNaughton and Papert 1971; Heinz 2010a; Rogers and Pullum 2011; Rogers et al. 2013). The set of forbidden substrings is denoted G , and their width, denoted k , is usually fixed for a given grammar/language. The subclass of SL with substring width k is called an SL- k grammar/language.

Example 1 Consider a natural language which exhibits progressive assimilation between a nasal and the following consonant. Licit words in such a language might include ‘ampa’, ‘anda’, and ‘amma’, but not ‘amta’ or ‘anma’. The forbidden substrings for this language are {mt, mn, np, nm}, making it SL-2. The licit word CVCV contains only the substrings {am, mp, pa}, none of which are forbidden. On the other hand, the illicit word ‘amta’ contains {am, mt, ta}, of which ‘mt’ is forbidden. This is summarized below.

(2) Example SL-2 pattern: local nasal assimilation

- $\Sigma = \{a, p, t, m, n\}$
- $G = \{mt, mn, np, nm\}$
- Licit words: ampa, anda, amma, anna, ...
- Illicit words: amta, anpa, anma, anma, ...

Following previous work, we will adopt the term k -FACTOR to refer to the length- k substrings of a word being modeled as well as the length- k strings that compose an SL- k grammar. This eliminates some terminological awkwardness, but more importantly, we will later generalize this idea to k -factors which are *not* identified with the substrings of a word, but other substructures such as subsequences or substrings of a tier projection.

We also want to be able to distinguish elements at the beginning or end of a word from those in the middle. To do this, we add the left and right edge markers $\bowtie$ and $\bowtie$ to each word, producing an ANCHORED WORD MODEL. So, *abc* becomes $\bowtie abc \bowtie$, *abcabc* is $\bowtie abcabc \bowtie$ and the empty string is $\bowtie \bowtie$. We can then make use of k -factors containing the edge markers in our SL grammars just like any other.²

Example 2 Consider a natural language which exhibits CV syllable structure with an optional word-final consonant. We can model this constraint as an SL language consisting of strings

²The older definition for an SL- k grammar (McNaughton and Papert 1971) lists forbidden factors at the beginning/middle/end of a word separately. The uses of anchored word models allows us to avoid this complication with minimal side effects. There is a small difference between these two approaches, which is that banning *ab* at the left edge of an unanchored word model is equivalent to banning $\bowtie ab$ in an anchored word model, which means that the older definition entails slightly greater expressive power. This can be corrected by reducing the factor size by 1 at the word edges.

of the symbols C and V, as summarized in (3). Licit words in this language include CV, CVC, and CVCV, but not VC, CVV, or CVCCV. The forbidden factors implementing the constraint are $\{\times V, VV, CC\}$, making it SL-2. The licit word $\times CVC \times$ contains the 2-factors $\{\times C, VC, VC, C \times\}$, none of which are forbidden. On the other hand, the illicit word $\times VC \times$ contains $\{\times V, VC, C \times\}$, of which $\times V$ is forbidden. For completeness, we also ban the empty string $\times \times$.

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

- $\Sigma = \{C, V\}$
- $G = \{\times V, VV, CC, \times \times\}$
- Licit words: $\times CV \times, \times CVC \times, \times CVCV \times, \times CVCVC \times, \dots$
- Illicit words: $\times V \times, \times VC \times, \times CVV \times, \times CVCCV \times, \dots$

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. $\perp$

Sidebar 2.2: Categorical and probabilistic grammars

Our k -factors are essentially categorical k -grams, making SL- k grammars the categorical equivalent of k -gram grammars (more commonly called n -gram grammars). The other classes discussed in this section all likewise have weighted and probabilistic equivalents. As such, everything that we do here with categorical grammars could in principle be adapted to probabilistic grammars as commonly used elsewhere in computational linguistics, and the same points about expressivity, locality and so on would still hold. Mayer (2021) defines a probabilistic version of tier projection and uses it to model gradient decay in long-distance phonotactics; Torres et al. (2023) extend this idea to model gradient in syntactic island effects. It is sometimes assumed that structure-focused approaches to language are inherently opposed to quantitative approaches, but this is simply not true.

From this point forward, we will generally omit the edge markers from words for brevity, but they are assumed to be implicitly present.

Now, the formal definition of an SL grammar/language almost exactly matches the informal definition, with one small complication. We need to make sure that we can handle (anchored) words with a length smaller than k . This is never a problem for SL-2 grammars, since even a word of length zero (the empty string) is really $\times \times$, but with SL-5, for example, we still want to be able to specify whether $\times ab \times$ is well-formed. We do this by including the whole word as a k -factor when its length is less than k .³

³This definition is based on those found in Rogers and Pullum (2011) and Rogers et al. (2013). An alternative approach seen in the literature is to add $(k - 1)$ edge markers to each end of the string, rather than just one.

Definition 1 (k -factors)

$$F_k(w) \stackrel{\text{def}}{=} \begin{cases} \{y : w = x \cdot y \cdot z, \ x, y, z \in \Sigma^*, \ |y| = k\} & \text{if } |w| > k \\ \{w\} & \text{otherwise} \end{cases}$$

Definition 2 A STRICTLY k -LOCAL (SL- k) grammar is a finite set $G \subseteq (\Sigma \cup \{\bowtie, \bowtie\})^k$.

Definition 3 A language $L \subseteq \Sigma^*$ is SL- k iff there exists an SL- k grammar G such that

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

Definition 4 an SL- k grammar/language is SL iff it is SL- k for some $k \geq 1$.

Equivalently, we can define an SL grammar as a set of PERMITTED factors. An SL grammar composed of forbidden factors is a NEGATIVE SL grammar, while the alternative based on permitted factors is a POSITIVE SL grammar. Converting between these two formulations is trivial, assuming that all substrings in the grammar have the same length and that well-formedness is categorical—we simply take the set complement w.r.t. the set of all possible k -factors. The main reason to prefer one version over the other is that either one may be more succinct depending on the SL language in question. When needed for clarity, I will use G^+ to indicate a positive grammar and G^- to indicate a negative grammar.

Definition 5 (Positive SL- k grammar) A language $L \subseteq \Sigma^*$ is SL- k iff there exists a positive SL- k grammar G^+ such that

$$L = \{w \in \Sigma^* : F_k(\bowtie w \bowtie) \subseteq G^+\}$$

Example 3 Consider again the example of CV syllable structure with an optional word-final consonant, now modeled with a positive SL-2 grammar as summarized in (4). Notice that G^+ is larger than G^- in this case, and that together they exhaust the possible 2-factors composed from $\{C, V, \bowtie, \bowtie\}$, excluding nonsensical combinations such as $\bowtie C$ and $\bowtie \bowtie$, which cannot occur in any string to begin with.

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

- $\Sigma = \{C, V\}$
- $G^- = \{\bowtie V, VV, CC, \bowtie \bowtie\}$
- $G^+ = \{\bowtie C, CV, VC, V\bowtie, C\bowtie\}$
- Licit words: $\bowtie CV \bowtie, \bowtie CVC \bowtie, \bowtie CVCV \bowtie, \bowtie CVCVC \bowtie, \dots$
- Illicit words: $\bowtie V \bowtie, \bowtie VC \bowtie, \bowtie CVV \bowtie, \bowtie CVCCV \bowtie, \dots$

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 SP and TSL) from more powerful classes in the subregular hierarchy. In contrast, if two symbols do occur within

the same k -factor, then we can make any arbitrary restriction on their mutual distribution, just as a finite grammar can make any arbitrary restrictions on the set of licit strings up to maximum length under consideration. For additional discussion of the formal properties of SL, the reader is referred to Rogers and Pullum (2011) and Rogers et al. (2013).

2.3.3 Strictly piecewise languages

In an SP language, the forbidden factors are SUBSEQUENCES of symbols in a word, which need not be contiguous, as opposed to substrings, which are contiguous (Heinz 2010a; Rogers et al. 2010). For example, the subsequences of *abca* are $\{aa, ab, ac, ba, bc, ca\}$.

Example 4 For our next example, consider a language with (symmetric) sibilant harmony, such as Samala (Heinz 2018). This is an SP-2 pattern, as shown in (5). Licit words in such a language include ‘saksa’ and ‘jakʃa’, but not ‘sakʃa’ or ‘jaksā’. We encode this by banning the factors sʃ and jʃ, so that harmony must hold no matter the distance.

(5) Example SP-2 pattern: Sibilant harmony

- $\Sigma = \{a, k, p, s, ʃ\}$
- $k = 2$
- $G = \{sʃ, jʃ\}$
- Licit words: asa, aʃa, saksa, jakʃa, sakapasa, ...
- Illicit words: saʃa, jaʃa, sakʃa, jaksā, jakapasa, ...

In this case, there is no reason to include edge markers, since every element is equally accessible to all the others. Also, it is conventional to define an SP- k grammar as including forbidden subsequences *up to* length k , so that SP-3 is a strict superclass of SP-2, just as SL-3 is a strict superclass of SL-2.

For our purposes, the key property of SP languages is observe is that while they are able to encode long-distance dependencies in a straightforward manner, they are completely *incapable* of handling blocking segments, or the equivalent in syntax.

Example 5 Consider a language with (symmetric) sibilant harmony which is blocked by other coronal obstruents, as in Slovenian (McMullin and Hansson 2016). In such a language, words like ‘sakataʃa’ are licit even though they disobey harmony in the strict sense.

This pattern is not SP- k for any k . For SP-2, some licit words contain the factors $\{sʃ, jʃ\}$, which means that they cannot be banned. One might try to move to SP-3, banning $\{skʃ, jks\}$ while allowing $\{stʃ, jts\}$, but then we would end up incorrectly ruling out ‘sakataʃa’, which contains the subsequence ‘skʃ’. The same problem occurs with larger k values. We can only ban disharmony completely, or allow it completely.

The problem with SP, as a linguist might describe it, is the complete and utter lack of locality restrictions. What we need instead is a model of long-distance dependencies which only allows certain elements to be skipped. This is exactly what TSL is.

2.3.4 Tier-based strictly local languages

A TSL language is much like an SL language, except that the forbidden factors are substrings on a tier. Any symbol not appearing on the tier is ignored completely; an SL grammar is the special case of TSL where every symbol appears on the tier (Heinz et al. 2011; Lambert and Rogers 2020).⁴ In this way, TSL languages allow a limited type of long-distance dependency to be expressed. And unlike SP, TSL can handle patterns involving blockers.

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 6 A TIER-BASED STRICTLY k -LOCAL (TSL- k) grammar is a tuple (T, G) where $T \subseteq \Sigma$ and $G \subseteq (T \cup \{\times, \bowtie\})^k$. 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 $k \geq 1$.

Example 6 Consider again the example of sibilant harmony with coronal blocking. (6). Licit words in such a language ‘sakapasa’ and ‘jakapafa’, as well as ‘sakatafa’, but not ‘sakapafa’. The tier alphabet contains only the sibilant consonants and the blocker. Mismatched sibilants are forbidden when adjacent on the tier, ruling out any strings that disobey harmony except when a non-sibilant coronal intervenes. For example, the illicit word sakapafa has the tier projection $\times s f \bowtie$, which contains the forbidden substring sf .

(6) Example TSL-2 pattern: Sibilant harmony w/ blocking

- $\Sigma = \{a, k, p, t, s, f\}$
- $T = \{s, f\}$
- $G = \{sf, fs\}$
- Licit words: sakapasa, jakapafa, sakatasa, jakatafa, sakatafa, jakatasa ...
- Illicit words: sakapafa, jakapasa, ...

While unbounded harmony is SP, and local spreading is SL, only TSL can handle long-distance harmony with blocking. In particular, the vast majority of attested harmony patterns are specifically TSL-2, as the presence of just a single blocker is always enough to break harmony (McMullin and Hansson 2016; McMullin 2016). In other words, there is no such thing as sibilant harmony which can skip one blocker but not two, up to two but not three, etc. This means that TSL-2 provides a good fit for the attested typology, in stark contrast to approaches based on optimization; see Heinz 2018 for additional arguments against optimization-based approaches to phonology.

Long-distance dependencies with blocking are particularly pervasive in syntax, making TSL-2 the prime candidate for the maximally restrictive classification of long-distance syntactic patterns. Graf (2022c) makes this point forcefully with respect to the attested types of

⁴TSL was inspired by autosegmental phonology (Goldsmith 1976), but is conceptually distinct, having no correlate of association lines between tiers.

island constraints, echoing the argument by McMullin and Hansson (2016). In this work, I make the same point regarding case and agreement.

2.3.5 Multi-tier grammars

The reader may notice that in (6) 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; Aksënova and Deshmukh 2018; Lambert 2025). 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 presented in Chapter 5 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 that interactions other than intersection are needed, in which case such patterns are not in fact MTSL. Due to the complexity of the data, a proper investigation of this issue is beyond the scope of the present study, which focuses primarily on “individual” dependencies.

2.3.6 Other parts of the subregular hierarchy

We have now seen three different ways in which the idea of forbidden factors can be implemented: over substrings, subsequences, and substrings on a tier, defining the classes SL, SP, and TSL, respectively. Of these, TSL provides an approximate upper bound for phonotactic dependencies, with the caveat that distinct tiers are needed for different dependencies, making MTSL the approximate upper bound for a complete grammar.

I should reiterate that there are exceptions to this generalization, with some patterns requiring a more powerful tier projection mechanism, yielding a structure-sensitive TSL (SS-TSL) language (Mayer and Major 2018; Graf and Mayer 2018; De Santo and Graf 2019). This was previously mentioned in Chapter 1. Interestingly, and more directly relevant to this work, there do not seem to be patterns which require (M)TSL- k with $k \geq 2$; if they are not (M)TSL-2 then they are SS-TSL, usually still with a window of 2. It is not clear what we should make of these facts, but they seem to me to be compatible with a learning-based explanation. Belth (2023, 2024) presents a learning model which posits a tier only when local constraints fail to be sufficiently predictive. One can imagine how if simple tier projection still fails, the learner might begin to posit more complex projection rules.

So SS-TSL is one direction of generalization that we can pursue. What about the others? At present, the abilities of other classes of the local-piecewise hierarchy presented in Section 2.2 (LT, PT, etc.) do not seem to necessary for natural language. To the extent that they once seemed relevant, TSL and its extensions have taken care of the attested patterns. Here too, a learning-based explanation might be appropriate, learning these other classes in full generality requires extraordinary amounts of memory (Lambert et al. 2021).⁵ Further work is required to determine whether this is indeed the crucial difference.

Other parts of the subregular hierarchy are more clearly relevant to human language. For example, Rogers and Lambert (2019b) show how forbidden factors can be generalized to other types of relational structures besides tiers. They also do the same for the complementary notion of MANDATORY factors, that is, factors that must be present for a structure to be licit. A good example of the latter is “a word must contain a stressed syllable”. By using such classes and their intersections, it is possible to account for the vast majority of known stress patterns. Going in the opposite direction, Lambert (2023) generalizes the idea of a tier to produce tier-based versions of other classes in the subregular hierarchy. Most of these are not, to my knowledge, linguistically relevant, but perhaps some could be. Putting together both of the above ideas, perhaps there are linguistic patterns which can be understood in terms of mandatory local factors on a tier.

For the purposes of this work, I only consider tree-based generalizations of the TSL languages, which in turn subsume the SL tree languages. It is entirely possible that some of the phenomena modeled in this way could also be handled with other subregular classes. Such an investigation is left to future work. What I can say is that none of the movement phenomena studied in this chapter, nor the case and agreement phenomena studied in subsequent chapters, require SS-TSL. I am also not aware of any syntactic phenomena which clearly do, binding being the chief exception.

This concludes our survey of subregular string languages. In the remainder of this chapter, we will apply the same general approach to the modeling of syntactic constraints over trees. It is now time to introduce the syntactic framework which provides the trees.

2.4 MG dependency trees

Minimalist Grammars (MGs, Stabler 1997, 2011) are a formalization of ideas from Chomsky’s (1995) Minimalist Program. In this section, I provide an informal overview of the MG feature calculus and the dependency tree representation of an MG derivation. Readers interested in the technical details of MGs are referred to Graf (2024a) and references therein.

For the moment, I restrict the discussion to a slightly simplified version of standard MGs as defined in Stabler (1997).⁶ A grammar is a LEXICON (set) of syntactic elements, each con-

⁵The same is actually true of MTSL, as the space of possible tier alphabets grows exponentially with respect to the size of the base alphabet. Then again, we do not expect that human learners actually entertain every possible tier alphabet. Other factors must be involved.

⁶The simplifications made in this section are as follows: semantic exponents are omitted, selector features which trigger head movement is omitted, and all licensor features follow all selector features. Also, the SMC is ignored here, but discussed in Section 2.5.

sisting of a phonetic exponent paired with an ordered string of FEATURES which control the Merge and Move operations, which implement structure building and displacement, respectively. The language of a grammar includes every structure which can be generated by combining lexical items in a series of Merge and Move steps which obeys the MG feature calculus.

The features for each operation come in two polarities. 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^- . 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^- (this is discussed in detail momentarily). 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. In short, in MG, *potential* movers do not bear a licensee feature, only *actual* movers do.

A few other points about MG bear mentioning at this point. First, every feature must be checked against a matching feature of the opposite polarity, with the exception of the category of the root node, which must be a member of a privileged set of ‘final’ categories. At a minimum, this set includes C^- so that the derivation can end at a full clause. By adding additional categories, fragments are also admitted. Related to the above, successive cyclic movement is not represented directly. If desired, traces or copies of moved phrases can be placed in intermediate landing sites as part of the derivation, or they can be omitted entirely. Either way, successive cyclic movement is not feature driven, avoiding many common problems. Finally, feature specifications are ordered, and grouped as follows: zero or more selector features, zero or more licensor features, exactly one category feature, and then zero or more licensee features.

MGs can generate a variety of structures. Here, we directly generate a language of DERIVATION TREES, which show the order of Merge and Move steps. The derivation tree can then be mapped to a phrase structure tree by executing all movements and inserting X' -style labels. Alternatively, it is possible to map the derivation tree to the corresponding morphological (and semantic) representations directly. Either way, 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 research; see Graf (2023) for a subregular model which handles placement of moved elements in the correct position.

The specific kind of derivation tree we will use is a DEPENDENCY TREE, in which every node represents a lexical item, and the subtree rooted at that node is the phrase headed by that item.⁷ An example phrase structure tree and the corresponding dependency tree for the sentence *The cat chases the rats* are shown in Figure 2.5. Every node in the dependency tree

⁷This precise type of MG dependency tree first appears in Graf and Shafiei (2019) and is formally defined in Graf and Kostyszyn (2021), but the use of dependency structures in MG has extensive precedent. The earliest use seems to be Kobele (2002), and those in Kobele (2012) are essentially identical to that used here. Also see Boston et al. (2010), who use MGs to derive surface dependency trees.

Sidebar 2.3: A note on notation

Here, I use F^+/F^- selector/category features and f^+/f^- for licensor/licensee features, following Graf (2018) and much of the subregular syntax literature. In Stabler’s (1997) original notation selector/category/licensor/licensee features are notated $=f/f/+f/-f$. In this work, we need an additional binary feature type for agreement, corresponding to the unvalued/valued distinction in Minimalist Syntax. Since different classes of features are easily distinguished by human readers on the basis of their labels, later in this chapter I adopt a simplified system which uses an undifferentiated $+F/-F$ for all of them.

Operation	Feature Type	Stabler (1997)	Graf (2018)	Simplified
Merge	Selector	$=f$	F^+	$+F$
	Category	f	F^-	$-F$
Move	Licensor	$+f$	f^+	$+F$
	Licensee	$-f$	f^-	$-F$
Agree	Unvalued	n/a	n/a	$+F$
	Valued	n/a	n/a	$-F$

is a lexical item, and the daughters of a node are its arguments in position of initial insertion, ordered in reverse merge order, that is, from right to left. 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 position only. In the present example, this applies to the subject, which undergoes EPP movement to Spec-TP.

This representation is extremely compact compared either to a phrase structure tree or a traditional MG derivation tree, in which the internal nodes represent the Merge and Move op-

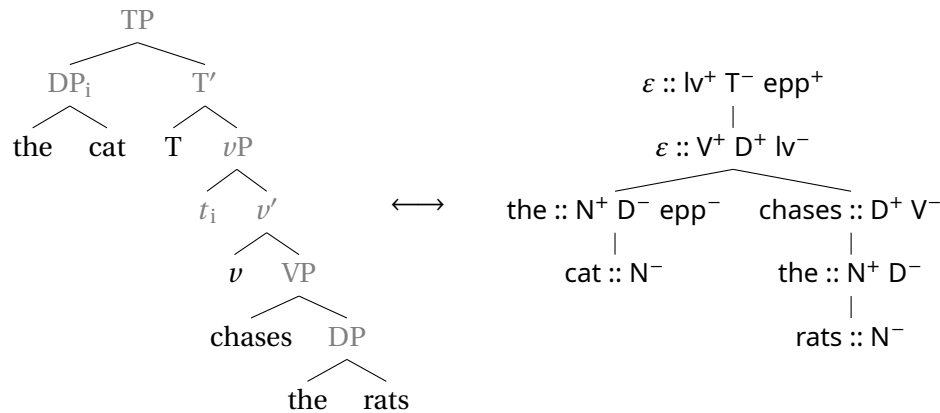


Figure 2.5: Phrase structure tree and dependency tree for “The cat chases the rats”. Nodes appearing only in the phrase structure tree are grayed out.

erations. Despite this, it contains everything we need to know in order to determine whether a derivation is well-formed. This is because the extra nodes appearing in other representations are completely determined by those in the dependency tree, and therefore contain no additional information. Also, note that while dependency trees are ordered, the ordering among sisters is a *structural* order, corresponding to c-command in the phrase structure tree. As such, this representation does not violate the principle that information about surface (pronounced) order is not a part of the syntactic representation.

There is yet another reason to prefer dependency trees which is particularly important to us: movement, agreement, and case are plausibly TSL over dependency trees, but SS-TSL is required if we use traditional derivation trees. Crucially, the extra power of structure sensitive tier projection is only needed to accommodate the excess nodes of the derivation tree. This increase in computational complexity can therefore be seen as a representational artifact, a remnant of the custom of identifying syntactic leaves/terminals with lexical items, dating to the context-free grammars used in the earliest versions of generative grammar.⁸

There are several linguistic aspects of the current example which also merit comment. In particular, some readers may be concerned about the EPP feature which triggers movement to Spec-TP. In the analysis I have in mind, every DP and CP (and perhaps certain PPs) can optionally bear epp^- , since these are the items which potentially undergo this type of movement. I have chosen this famously meaningless label in part because it does not imply any particular theory of what drives said movement, whether it be case assignment, ϕ -agreement, or anything else. Ultimately, languages differ in which elements undergo which types of movement, making this a point of parametric variation like any other. In general, there is no significance to the fact that language X has value Y for parameter Z (at least, not in synchronic terms). What we can and should explain is the overall space of possibilities.

So, for the purposes of this dissertation, I adopt the above analysis of EPP movement, which is completely parallel to the standard MG treatment of *wh*-movement. In each case, we have a class of landing sites, which obligatorily bear the licenser feature, and a class of movers, which optionally bear the corresponding licensee feature. We also need to constrain the structural distribution of these features, so that relativized minimality and other general constraints are enforced. Furthermore, we must account for exceptions to the generalizations, such as the ability for d-linked *wh*-phrases to subvert minimality, as well as parametric variation, such as the difference between single and multiple *wh*-movement, and the same can be said for EPP movement as well.⁹ In the remainder of this chapter, I show how TSL grammars can do this. For reasons of space, I provide only a brief sketch of the constraints on movement, since our focus is ultimately on case and agreement, and many of the same principles apply across the board.

⁸In contrast to phonology, in which there are a handful of phenomena which seem to require structure sensitive tier projection, in syntax there are no known examples aside from binding (Graf and Shafiei 2019), which is notoriously complex in many ways, and may not even be syntactic.

⁹For example, Mikkelsen (2005) uses a ‘Topic’ feature to derive inversion of the subject and predicate in copula clause constructions. Essentially, it creates a more specific type of movement which subverts relativized minimality in the same way that d-linking subverts it for *wh*-movement.

Sidebar 2.4: Making sense of MG

Without further restrictions, the MG feature system could also be used to construct a language where, for example, some DPs are optional EPP-movers, some are obligatory movers, and some are non-movers, each group completely arbitrary. This and other aspects of MG might strike syntacticians as bizarre, but this is simply because MG is not a complete syntactic theory. It is rather a formal framework with approximately the right computational properties, within which a complete theory can be built (the same can be said of other formalisms such as TAG and CCG). Subregular syntax is another component of such a theory, narrowing down the space of possible derivations in a principled manner. We also need a theory of the MG lexicon, and many other components besides these. That MGs enable such a modular approach to theory building is a major advantage.

2.5 Constraining the derivation

Standard MGs generate regular derivation/dependency tree languages, and in combination with a regular tree-to-tree map (enacting movement) and a linearization procedure, generate string languages which are MCFL. In light of the preceding discussion, we can now state our goal as follows: we wish to show that the constraints on the dependency tree language needed for natural language are TSL. In general, each type of dependency requires a separate tier, which means that the tree language as a whole is MTSL. Though our aims are conceptually simple, there are a number of technical considerations which must be sorted out.

First, there are several ways in which TSL string languages have been extended to trees, which can be broadly split into two categories. Graf (2018), Vu et al. (2019), Graf and Kostyszyn (2021), and Graf (2022c) employ a direct tree-based analog of string TSL: the tiers are trees, and the constraints restrict the string of daughters of each node on the tier.¹⁰ 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 will refer to the former approach as *TREE TSL* or *TREE TIERS*, and the latter as *PATH TSL* or *TIERS OVER PATHS*. Neither approach is capable of handling all syntactic constraints. Tree TSL cannot enforce the c-command constraint, while path TSL cannot coordinate information across distinct paths. At present, it seems that both systems are necessary, and I utilize both in this thesis. Tree TSL is covered in Section 2.6, and path TSL is covered in Section 2.7.

Next, we must ensure that the basic rules of the MG feature calculus can be stated as TSL constraints over dependency trees. Assuming the standard MG feature system, the principle constraints on well-formed trees can be summarized as follows:

1. **MERGE**: Every selector feature on a non-root node must be paired with a single corre-

¹⁰Graf (2018) and Vu et al. (2019) utilize traditional derivation trees, and therefore require SS-TSL. Later work, starting with Graf and Kostyszyn (2021), use dependency trees, in which TSL is sufficient for most syntactic phenomena.

sponding category feature, and vice versa.

2. FINAL: The category feature of the root must be a member of the permissible set of final categories.
3. MOVE: Every licensor feature must be paired with a single corresponding licensee feature, and vice versa.
4. SMC: Only one instance of each distinct licensee feature may be active at any point in the derivation.

For MERGE, we can confirm that all selectional restrictions are satisfied by looking just at the selector features of a node and the category features of its children. This means that MERGE is SL-2. The same can be said of FINAL, which only requires us to look at the root node, the sole child of the top edge marker. There is a small complication with this claim, which is that recursive adjunction can potentially introduce non-locality for selection (Graf 2018). Although I do not deal much with selection or adjunction in this work, I assume that adjuncts are daughters of the head, just like arguments, in which case both selection and adjunction are SL, inclusive of recursive adjunction (Hanson 2025a).

For MOVE, we must match up licensor features with their corresponding licensee features. In this case there is no finite bound on the distance, so Move is necessarily (at least) TSL. This brings us to the SMC. SMC stands for ‘shortest movement constraint’, though it is quite different from the constraint from Minimalist Syntax which shares its name. It exists to guarantee that the derivation is deterministic: if at any point there are two movers which might be selected to move to a given landing site, the derivation is simply aborted. Both MOVE and SMC are tricky to calculate for standard MGs since licensee features are ordered, and only become active as preceding features are checked. Graf (2022c) resolves this by stipulating that licensee features are *unordered* on lexical items; the order of movement steps is therefore determined solely by the mover’s *set* of licensee features in combination with the derivational context in which the item is placed. In this case, both MOVE and SMC are TSL, specifically using tree tiers. Importantly, strong and weak generative capacity are also preserved. Constraints on movement are treated in Section 2.6.

What about other syntactic constraints, such as relativized minimality? While the aforementioned constraints can be handled with tree tiers, relativized minimality cannot, because tree tiers do not preserve c-command relations from the full tree. Instead, we must use PATH TSL, as discussed in 2.7. The paths are defined so that string order encodes c-command, roughly speaking, and projecting a tier from those paths preserves this relation.

Now, most of the above constraints, as stated, are not quite empirically adequate. They represent a reasonable baseline, but with well-documented exceptions. Other constraints, such as islands, do not have anything to do with the MG feature calculus. Our solution is allow the constraints on movement dependencies to vary somewhat across languages, and even between different movement types within the same language, as long as they remain TSL. Once we make this step, extending the approach to other dependencies, such as adjunction, case, and agreement, is trivial. We simply introduce additional features, and constrain their distribution using TSL grammars. This is the focus of the following chapters.

The resulting system, which we might call a TSL MG, is somewhat different from what we

started with. Whereas MGs (both standard and extended) usually place all cross-linguistic variation in the lexicon, this is no longer the case. This is innocuous from a mathematical perspective, since (leaving aside some minor technical details) every regular tree language can be translated into a standard MG lexicon, and vice versa.¹¹ The current work is ultimately built on results in model-theoretic syntax that take advantage of this perspective on MGs; see Graf (2013) and references therein. Also see Graf et al. (2016) regarding the relation between TSL MGs and standard MGs, and Rogers (1998) for a model-theoretic implementation of a large fragment of Government and Binding Theory.

2.6 Tree tiers

In this section, we overview TSL tree languages as presented in Graf and Kostyszyn (2021) and Graf (2022c), and use them to model constraints on movement and case. We also discuss some of the basic predictions of the model.

2.6.1 Overview

Tree TSL operates in two logical steps. First, a tree tier is constructed in a manner which is exactly analogous to a string tier. We specify a tier alphabet and project a tier by removing nodes with any label not in this set, preserving the dominance and precedence relations. Second, each node on the tier then imposes constraints on its string of daughters with a TSL language determined the label of the mother. If every such daughter string is licit, then the tree tier is licit, and so is the original tree.

We begin with the tree tier. A toy example tree with nodes labeled $a/b/c/x$ and several tier projections are given in Figure 2.6, illustrating several important formal properties. The top and bottom edge markers are included in each tier projection, and the edges are dotted to emphasize that some items in the full tree may be skipped. The first tier projection, with $T = \{a, b\}$, shows how nodes which are distant in the original tree become local on the tier. Node a is now the mother of all nodes labeled b , which in turn are now sisters. The second tier, with $T = \{c\}$, shows how the maximum branching factor can be greater than that of any node in the original tree. Since there is in general no finite bound on the size of the original tree, the branching factor on the tier is also unbounded, even if it is bounded in the original tree language. The third, with $T = \{x\}$, shows tier projection that would multiple roots if not for the top edge marker, making this crucial in maintaining a tree representation.

Next are the constraints. Consider a TSL tree language whose tree tier consists of the elements $\{a, b\}$ with the following constraints: the daughters of each a include least one b , and every b has a daughter string in which all a s precede all b , with no other restrictions. This is summarized in (7). The tree in the previous example satisfies these constraints. Additional examples are given in Figure 2.7.

¹¹Since the MG lexicon is not a theory of the human lexicon, no particular psychological interpretations should be drawn. Nonetheless, there are linguistically interesting issues involving the relationship between lexical items and their features, on the one hand, and tree constraints, on the other. See Graf (2017a) for details.

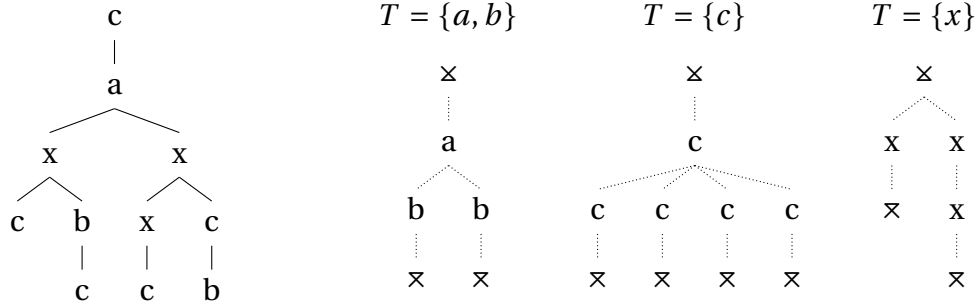


Figure 2.6: Tree and three different tiers, demonstrating various formal properties.

(7) Example tree TSL grammar

- $T = \{a, b\}$
- Daughter string languages:

Tier mother	Daughter string	Complexity	Grammar
a	$\Sigma^* b \Sigma^*$	TSL-2	$T = \{b\}, G^- = \{\times \times\}$
b	$a^* b^*$	SL-2	$G^- = \{ba\}$
$\times$	Σ^*	SL-1	$G^- = \{\}$

The daughter string languages are TSL-2, SL-2, and SL-1 respectively, with TSL-2 being the upper bound. The tree tier itself can be said to be TSL-2, since all constraints consider only a node and its daughters on the tier. In Graf’s (2022) classification, we would therefore categorize this TSL tree grammar as a TSL-2[TSL-2] grammar. If we weakened the daughter string language of a to be SL-2, then we would have a TSL-2[SL-2] grammar. To the extent that tree tiers are an effective model of syntactic constraints, TSL-2[TSL-2] appears to be the upper bound. That is, we do not need SS-TSL or TSL-3 in either step.¹²

2.6.2 Pairing movers and landing sites

Recall that (by default) movement step must have exactly one mover and landing site, which must bearing matching licensee and licensor features. Also, we are assuming that licensee features are unordered, so items which move multiple times do not get to decide the order of movement types, but rather must take the closest landing site for each. So, for example, a DP which bears both epp^- and wh^- must undergo EPP-movement followed by wh -movement, not because the features are ordered, but rather because the T head bearing epp^+ is always closer than the C head bearing wh^+ .¹³

¹²Actually, TSL-3 for tree tiers has yet to be defined. It is unclear what this would even look like. Are the nodes at depth 3 treated together, or separately according to their mother? Is their string language determined by the mother in combination with the aunts and grandmother, and if so, how?

¹³Even though licensee features are unordered, there may be a consistent order among the movement steps themselves; I have found this to be true of the MGBank corpus Torr (2017). If generally true, the existence of a language-wide order actually obviates the need to order the features of individual movers. It also helps with determining the final landing site when implementing movement as a TSL tree transduction (Graf 2023).

We construct our TSL grammar as follows. For each movement type F , we project a tier which contains all elements which bear either f^+ or f^- . On this tier, we require that a) every item bearing f^+ have *exactly one* element bearing f^- among its daughters, and b) every element bearing f^- have a mother bearing f^+ , that is, it cannot appear in the daughter string of any other element. We can summarize this as follows:

(8) TSL tree grammar for pairing movers and landing sites of type F

- **Tier:** Project all X in LEX bearing either of $\{f^+, f^-\}$
- **Constraints:** Let l be a landing site for feature F and m be a mover. The daughter string language $ds(x)$ for each element x is:

$$ds(l) := \overline{m}^* m \overline{m}^*$$

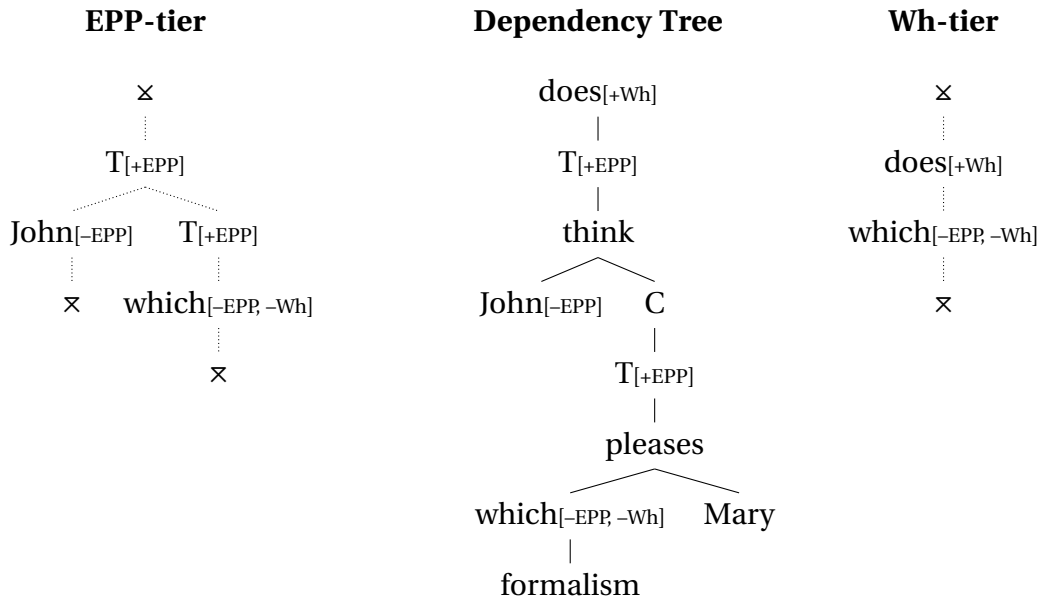
$$ds(m) := \overline{m}^*$$

$$ds(\times) := \overline{m}^*$$

For ease of readability, I now introduce a simplified notation for lexical items (this was briefly mentioned in Section 2.4). Each item is labeled $X_{[F, G, H, \dots]}$, where X is the phonetic exponent or, for empty categories, a descriptive label, and $F/G/H/etc.$ are MG features. We omit all category and selector features when not directly relevant, including just those involved in movement (and later, case and agreement). No ordering among licensee features is implied. Licensor features are ordered only implicitly, and we will not treat any examples with multiple licensor features in any case.

Figure 9 shows an example with both EPP-movement and *wh*-movement, reproduced from Graf (2022c) with slight modifications. On the EPP-tier, each finite T has one EPP-mover among its daughters, and on the *Wh*-tier interrogative C has one *wh*-mover, as required.

(9) Dependency tree and tiers for *What formalism does John think pleases Mary.*



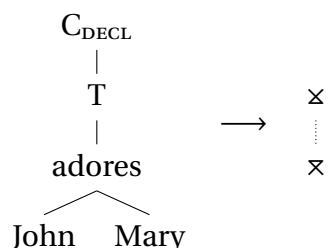
Let us focus just on *wh*-movement for the moment, and consider some ways in which our TSL grammar could be violated. There are three basic cases:

1. $X_{[+F]}$ has no $X_{[-F]}$ as a daughter
2. $X_{[+F]}$ has more than one $X_{[-F]}$ as a daughter
3. $X_{[-F]}$ has no $X_{[+F]}$ as a mother

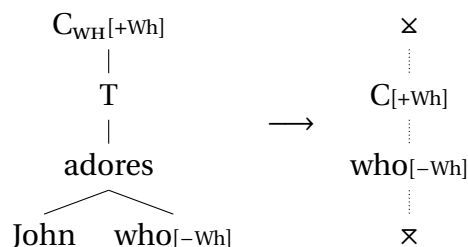
Case 1 corresponds to a *wh*-question in which no *wh*-phrase moves (an *wh-in-situ*), or no such phrase exists (which is nonsensical). Case 2 corresponds to multiple *wh*-movement, which is not allowed in English but would be allowed in Russian. Case 3 corresponds to a *wh*-mover with no landing site (also nonsensical). All of these are illustrated in (10).

(10) Good and bad *wh*-movement configurations in English (simplified structures)

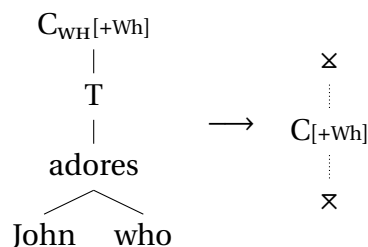
- a. ✓ Declarative clause, cf. *John adores Mary*



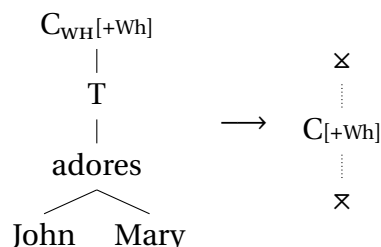
- b. ✓ Single *wh*-movement, cf. *Who does John adore?*



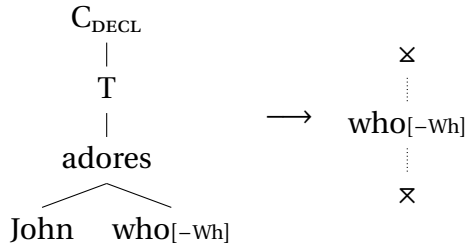
- c. ✗ *wh*-in-situ, cf. *John adores who?*



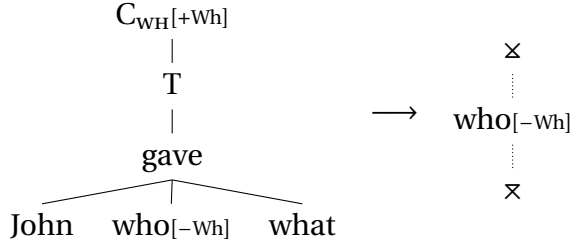
- d. ✗ No *wh*-mover (incoherent)



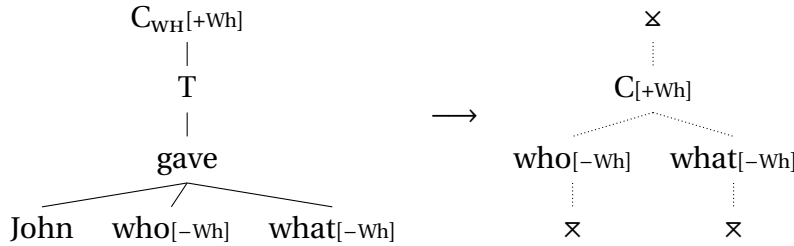
- e. ✗ No *wh*-landing site (incoherent)



- f. ✓ Multiple *wh*, single movement, cf. Who did John gave what?



- g. ✗ Multiple *wh*, multiple movement, cf. Who what John gave?



There are many other constraints that are not (yet) represented in our grammar. At the same time, it already holds the seeds of syntactic locality, since if any other elements were to project on the tier, they could separate a mover from its landing site. We discuss these next.

2.6.3 Variation in the tier projection

Different movement types are well-known to have different locality profiles, though there is much disagreement in the literature on how best to hand this variation. In the TSL syntax literature, locality conditions are handled by treating certain elements as BLOCKERS ON A TIER, which is exactly analogous to the treatment of TSL treatment of blocking in long-distance harmony which was shown in Section 2.3.4.

For example, EPP-movement is bounded by full CPs in English, though *wh*-movement is not. We illustrate this by juxtaposing a non-finite raising structure, assumed to involve an embedded TP, and the (impossible) hyperraising structure, with raising from embedded CP. Recall that in MG, successive cyclic movement is not represented; the only gap therefore appears in the base position of the moved DP.

- (11) No EPP-movement across CP

- a. We seem [_{TP} to ____ have a problem].

- b. * We seem [_{CP} (that) have ____ a problem].

In contrast, we can *wh*-move a DP out of both structures, whether or not the *wh*-DP originated as a subject.

- (12) *Wh*-movement of embedded subject permitted across CP

- a. Who seems [_{TP} to ____ have a problem]?
b. Who does it seem [_{CP} has ____ a problem]?

- (13) *Wh*-movement of embedded object permitted across CP

- a. What do we seem [_{TP} to have ____]?
b. What does it seem [_{CP} (that) we have ____]?

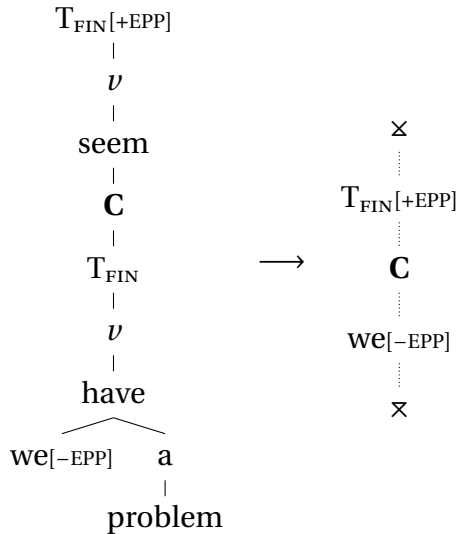
In the terminology of Keine (2019, 2020), C is a HORIZON for EPP movement, but not for *wh*-movement. (We discuss horizons briefly below.) In the TSL analysis, declarative C projects on the EPP-tier, but not on the Wh-tier. This means that in addition to movers and landing sites, the tier for each grammar also includes any blockers for said movement type.

- (14) EPP and *wh*-movement in English (provisional)

- a. Tier for EPP movement: {all T_{FIN}[+EPP], all D[-EPP], C_{DECL}}
b. Tier for *wh*-movement: {all C_{WH}[+Wh], all X[-Wh]}

When we project the tier for EPP-movement for the hyperraising example, the mover and landing site will be separated by the complementizer. As a result, the constraint requiring strict pairing of movers and landing sites is violated, and the structure is correctly ruled out.¹⁴

- (15) Blocking of EPP-movement by C (ex. 11b)



¹⁴For the purposes of this example, we assume that embedded T need not bear the +EPP diacritic, as successive cyclic movement is not conventionally represented in MG. If it did bear the diacritic, then this alone would result in blocking, based on our current grammar. In principle, ensuring that only the final landing site bears the diacritic requires a separate tier which contains all instances of T_{FIN}. If this seems distasteful, the chain approach to iterated agreement in Chapter 7 might be used in concert with overtly represented successive cyclic movement. I leave an exploration of these ideas to future work.

Now, just because *wh*-movement is not sensitive to declarative CPs does not mean that it is completely unrestricted. On the contrary, interrogative CPs *do* block *wh*-movement, not to mention EPP-movement. Similarly, *wh*-movement is impossible out of relative clauses, clausal complements of nouns, and complements of many verbs.

(16) Blockers for *wh*-movement in English

- a. *Who do you wonder [_{wh-CP} whether John adores ____]?
- b. * Who did Fred adore the cat [_{RC} which adores ____]?
- c. * Who did Fred believe the claim [_{NComp} that John adores ____]?
- d. * Who did Fred whisper [_{VComp} that John adores ____]?

These are just examples of ISLAND CONSTRAINTS (Ross 1967). Under a TSL analysis, many (though not all) islands are also just blockers on a tier. The difference between a horizon and an island is somewhat subtle. Horizons are claimed to include all members of a certain functional category, while islands are somewhat more arbitrary. Furthermore, Keine's theory of horizons attempts to explain which categories may or may not be horizons for movement targeting a particular landing site, while variation in islands seems to be more related to properties of the precise subvariety of movement (e.g. movement of so-called d-linked *wh*-phrases out of island structures is judged more favorably than movement of simplex *wh*-phrases). While these two types of blockers may be conceptually distinct, they are identical in their behavior: a single horizon or island rules out movement of the relevant type. We therefore unify them formally as blockers on a tier.

(17) EPP and *wh*-movement in English (revised)

- a. Tier for EPP movement: {all T_{FIN}[+EPP], all D[−EPP], all C}
- b. Tier for *wh*-movement:
 {all C_{WH}[+Wh], all X[−Wh], whether, C_{REL}, claim[−C], whisper[−C], ...}

Now, this treatment of syntactic blocking makes it seem as if almost anything can potentially serve as a blocker. Indeed, we predict that various types of elements can serve as blockers for different dependencies, some of which may seem quite inexplicable in terms of any grammatical principle. From this perspective, a blocker is essentially an element that is inactive for a given dependency yet nonetheless projects on its tier, and in a sense 'should not be there'. Furthermore, we predict that elements which block one dependency do not necessarily block others, and that the exact state of affairs differs somewhat between languages. Given our current knowledge, all of this is true: both horizons and islands vary in their distribution across individual dependencies and languages. Therefore, the fact that element X is a blocker for dependency Y in language Z *cannot* generally be derived from any deep principle. This is not to say that there are no further generalizations to be made, but it seems most likely that they will be tendencies rather than strict rules, and that they are based in factors which are not purely grammatical in origin.

2.6.4 Variation in the daughter string language

In the basic TSL analysis of MG-style movement just outlined, we make use of a TSL-2 string language to enforce the constraint *exactly one*. This can be decomposed into the constraints *at least one* and *at most one*, each of which is also TSL-2. As argued by Graf (2022c), these also correspond to actual movement phenomena: multiple movers to a single landing site, and optional movement. Another logical possibility is an item which acts like it bears +F and –F simultaneously, which corresponds to items which exist only along a movement path of the right type.

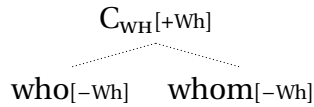
To begin, multiple *wh*-movement is well-attested in many Slavic languages. For example:

- (18) Multiple *wh*-movement in Bulgarian (Bošković, 2002, p.353)

[Ko_i koga_j [t_i voli t_j]]?
 who whom loves
 ‘Who loves whom?’

Abstracting away from details such as superiority effects,¹⁵ this is handled by relaxing the *at most one* sub-component of the constraint on single movement, allowing a landing site to have multiple daughters of the right type.

- (19) Example Wh-tier for multiple *wh*-movement



To enact this, the grammar of the daughter string language need only permit the bigram $X[-Wh] \cdot X[-Wh]$.

- (20) Basic grammar for multiple movement

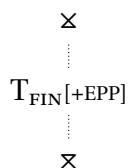
- Project: all $X_{[+F]}$, all $X_{[-F]}$
- Constraints: daughters of $X_{[+F]}$ include at least one $X_{[-F]}$
 - $T = \{X_{[-F]}\}$
 - $G^+ = \{\bowtie \cdot X_{[-F]}, \textcolor{blue}{X}_{[-F]} \cdot \textcolor{blue}{X}_{[-F]}, X_{[-F]} \cdot \bowtie \}$

For optional movement, in contrast, we relax the constraint to *at most one*. Graf (2022c) uses this to analyze option insertion of *there*, taking *there* to be the spell-out of T_{FIN} which failed to find an EPP-mover.

- (21) a. A man is in the garden.
 b. There is a man in the garden.

¹⁵To simplify somewhat, in some but not all languages with multiple *wh*-movement, highest of the moving phrases must also be highest in the post-movement order, even if the order of the remainder varies. In others, any order is allowed.

(22) EPP-tier for (21b) under Graf’s (2023) analysis



In this case, we add the bigram $\times \cdot \times$ to the grammar.

(23) Basic grammar for optional movement

- Project: all $X_{[+F]}$, all $X_{[-F]}$
- Constraints: daughters of $X_{[+F]}$ include at most one $X_{[-F]}$
 - $T = \{X_{[-F]}\}$
 - $G^+ = \{\times \cdot X_{[-F]}, X_{[-F]} \cdot \times, \times \cdot \times\}$

Of course, EPP-movement is normally obligatory in English, so more needs to be said to get this particular analysis off the ground. Actually, expletive *it* might be more amenable to such an analysis, since it appears whenever a CP fails to move, whereas *there* is only available with specific lexical predicates, suggesting that it is selected and moves like any other DP (see Deal 2009 for additional arguments for a movement-based account of existential *there*). But such an analysis could be appropriate in a language where a given movement type is always optional.

2.6.5 Impossible islands

Mirroring the argument by McMullin and Hansson (2016) that TSL-2 gives the right set of locality profiles for long-distance harmony, Graf (2022c) imagines a number of logically possible yet unattested types of islands. I repeat his list below, as it is extremely illustrative:

(24) Impossible island types (Graf 2022c, p. 7)

- a. **Gang-up island effects**
A mover can escape n islands, but not $n + 1$.
- b. **Configurational island effects**
XP is an island iff it is inside an embedded clause.
- c. **Cowardly island effects**
XP is an island iff there are at least n XPs in the same clause.
- d. **Narcissist island effects**
XP is an island iff there are no other XPs in the same clause.
- e. **Rationed island effects**
At most n phrases per clause can be an island.
- f. **Discerning islands**
XP is an island only for movers that contain a PP.

Although not stated explicitly in Graf (2022c), all of these constraints are first-order definable, and therefore subregular. But even though they are not very complex, they are completely unlike anything found in natural language. What they all have in common is that they are not TSL-2. This means that they can be ruled out on purely formal grounds.

To briefly summarize, with tree tiers and a window of two, we can achieve a tight fit for the kinds of constraints which characterize syntactic movement, while ruling out many other patterns which do not exist. Next, let us consider a somewhat different implementation of TSL-2 in syntax.

2.7 Tiers over paths

In this section, I introduce two path-based models: `COMMAND STRINGS` (Graf and Shafiei 2019) and `ANCESTOR STRINGS` (Shafiei and Graf 2020). Rather than projecting a tier from the tree directly, we extract a set of strings representing the `PATHS` along which syntactic dependencies occur. The set of all licit paths is a `PATH LANGUAGE`, which we define using a TSL string grammar. The reason to do this is partly for mathematical convenience, since we know that our path languages can be compiled into a tree grammar which implements them directly (Graf and De Santo 2019). But it is also partly linguistic, for syntactic dependencies tend not to hold between items that are deeply embedded in distinct subtrees. This can be easily enforced by requiring that our paths always follow the direction of the dominance and left-sister relations.

2.7.1 Ancestor strings

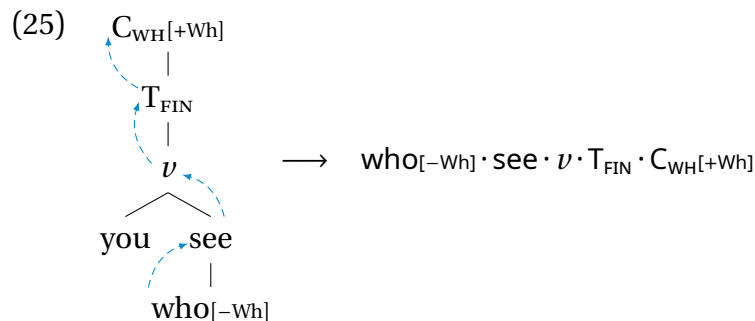
We begin with what might be the simplest useful path-based model: `ANCESTOR STRINGS`, or `A-STRINGS` (Shafiei and Graf 2020). The a-string of a node is just its string of ancestors (plus itself) in reverse dominance order. The formal definition is given below.¹⁶

Definition 7 (a-string) Let t be an MG dependency tree, let l be the node labeling function, and let $\cdot$ be the concatenation operator. For every node n in t , its a-string $as(n)$ is defined recursively as follows:

$$as(n) \stackrel{\text{def}}{=} \begin{cases} l(n) & \text{if } n \text{ is the root of } t \\ l(n) \cdot as(m) & \text{if } m \text{ is the mother of } n \end{cases}$$

For example, the a-string of ‘who’ in *Who did you see?* is $\text{who} \cdot \text{see} \cdot \nu \cdot T_{\text{FIN}} \cdot C_{\text{WH}}$.

¹⁶I have altered the original definition from Shafiei and Graf (2020) to more directly parallel the definition of a c-string. Additionally, it has been made more precise by putting the label of a node in the string rather than the node itself, which in the axiomatization assumed here is just a Gorn address.



It is easy to see how a-strings can be used to model certain island constraints using a TSL-2 string grammar. As with tree tiers, we simply project all movers, landing sites, and blockers, and require that each landing site and mover are adjacent on the tier (in the correct order).

(26) TSL-2 grammar for selected *wh*-islands

- Tier: {all $C_{\text{WH}[+Wh]}$, all $X_{[-Wh]}$, whether, C_{REL} , $\text{claim}_{[-C]}$, $\text{whisper}_{[-C]}$, ...}
- Constraints: $\{X_{[-Wh]} \cdot Y, X_{[-Wh]} \cdot \bowtie, X \cdot Y_{[+Wh]}, \bowtie \cdot Y_{[+Wh]}\}$

In general, a-string TSL-2 grammars are a strict subclass of tree TSL-2 grammars, as they lack any information about tier sisterhood. This being the case, they cannot enforce any constraints that require such information, such as the restriction of one mover per landing site.¹⁷ Therefore, they function primarily for analytical convenience when treating constraints for which the full power of tree tiers is not needed. This is not the case for command strings, as we will soon see.

Sidebar 2.5: More on islands with a-strings

Shafiei and Graf (2020) do not actually use TSL in their model of island constraints, but a more powerful class of string languages known as IBSP (Graf 2017b). As the typological naturalness of the TSL-2 languages as a model of syntactic locality has become more apparent since their work, the appeal of IBSP in this context has diminished somewhat.

2.7.2 D-command and command strings

COMMAND STRINGS, or c-strings, were first introduced by Graf and Shafiei (2019) as a means to model constraints such as NPI-licensing and binding, which are conventionally described in terms of the c-command relation (Reinhart 1981) within a phrase structure tree. In doing so, they introduce a new relation, DERIVATIONAL COMMAND or d-command, which is appropriate to the structure of a dependency tree.

¹⁷A-string are also subsumed by command string grammars if the latter are augmented to distinguish containing and non-containing nodes, as in De Santo and Graf (2019). To my knowledge, this technique has not been utilized in any linguistic analysis yet.

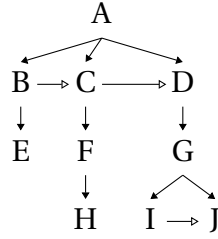


Figure 2.8: D-command as a strict partial order. Solid arrows represent dominance, and dotted arrows represent left-sister. Every directed path between two node represents d-command. For example, node C d-commands E, H, and the entire subtree rooted at D. The root d-commands everything, and the leaves d-command nothing.

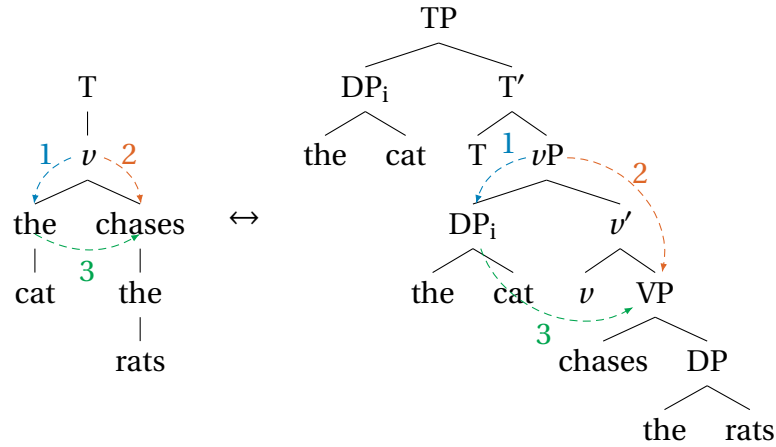


Figure 2.9: D-command in the dependency tree takes the place of both c-command and m-command in the phrase structure tree. v m-commands both its specifier and complement (it also c-commands the complement), and the specifier m-commands the complement.

The the conventional definition for c-command (X c-commands Y iff $Y \neq X$ the closest node Z dominating X also dominates Y) will not work for a dependency tree, as sisterhood here is an inherently asymmetric relation, while c-command among sisters is symmetric. Furthermore, it is almost always asymmetric c-command relations in the phrase structure tree (X c-commands Y , but Y does not c-command X) that are of linguistic relevance. We therefore define d-command to be asymmetric from the start.

Definition 8 (d-command) For any nodes x and y in tree t , x d-commands y iff x dominates y or x is the left sister of a node z which dominates y .

In effect, we have combined dominance and left-sister into a single ordering relation. This is visualized in Figure 2.8. Better still, it is a STRICT PARTIAL ORDER, just like (proper) dominance and left-sister. This means that d-command is very natural in mathematical terms, which in turn makes the definition of a command string extremely straightforward. In contrast, neither symmetric nor asymmetric c-command is a partial order (strict or weak), since they are not transitive.

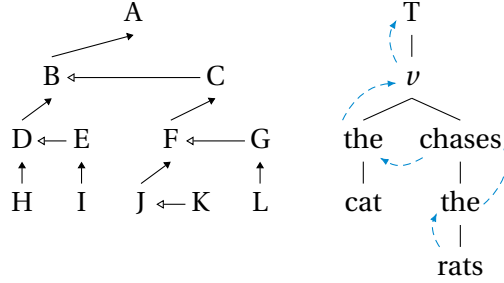


Figure 2.10: Left: a c-string follows the mother-of relation to the first daughter (filled arrow) and the left-sister relation (open arrow). The mother-of relation to non-first daughters (dotted lines) is not used. Right: c-string for *rats* in the sentence ‘The cat chases the rats’ (blue dashed lines)

In linguistic terms, d-command takes the role of asymmetric c-command, since the complement is commanded by both its head and the specifier, as well as m-command (Aoun and Sportiche 1982), since the head commands the specifier rather than the other way around. For example, ν d-commands its complement VP as well as its DP specifier, and DP also d-commands VP. This is illustrated in Figure 2.9. Also notice that, since d-command is defined in terms of the dependency tree, it collapses the distinctions between projections of a head, avoiding several complications. 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 superior to the specifier. D-command provides the latter, as desired, since it is the position of the maximal projection which is directly encoded in the dependency tree.

This brings us to c-strings. Just as an a-string encodes ancestors of a node in reverse order of dominance, the c-string of a node encodes its d-commanders in reverse order of command. The formal definition of is given below.¹⁸

Definition 9 (c-string) Let t be an MG dependency tree, and let l be the node labeling function. For every node n in t , its c-string $cs(n)$ is defined recursively as follows:

$$cs(n) \stackrel{\text{def}}{=} \begin{cases} l(n) & \text{if } n \text{ is the root of } T \\ l(n) \cdot cs(m) & \text{if } m \text{ is } n\text{'s mother and } n \text{ has no left sibling} \\ l(n) \cdot cs(s) & \text{if } s \text{ is } n\text{'s left sibling} \end{cases}$$

The set of c-strings can be visualized similarly as follows. We zigzag up the tree from the starting node, always going as far left as possible before going up. This is illustrated on the left side of Figure 2.10. As a concrete example, the c-string for *rats* in our running example sentence is *rats* · *the* · *chases* · *the* · ν · T, as illustrated on the right side of the same figure.

For a simple but illustrative application, I model a simplified version of the licensing condition for NPIs (negative polarity items) such as ‘anything’, ‘anyone’, ‘anywhere’, etc., which can only appear in the domain of a select set of licensors, including but not limited to neg-

¹⁸I have altered the original definition to make it easier to parse, and as with the definition of an a-string, corrected the definition to use the label of the node rather than the node itself.

ative elements. As a first order approximation, an NPI must be c-commanded—or rather, d-commanded—by such an element.

(27) NPI licensing in English

- a. * I saw **anything**. (no licensor)
- b. I did not see **anything**. (commanded by sentential negation)
- c. No one saw **anything**. (commanded by negative DP)
- d. * **Anyone** saw nothing. (no commanding licensor)
- e. * [A picture of no one] surprised **anyone**. (no commanding licensor)

Like many other syntactic dependencies, NPI licensing is unbounded and subject to blockers: while it can cross DP and CP boundaries, definite DPs tend to block licensing.

- (28) a. I didn't think [**that** she found [**a** picture of anyone]].
 b. * I didn't think [**that** she found [**the** picture of anyone]].

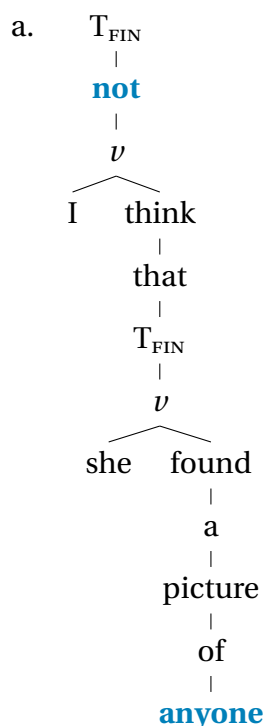
This is highly suggestive of a TSL-2 analysis, shown below. The first constraint handles cases of blocking, while the second handles the case where no licensor can be found.

(29) C-string grammar for NPIs

- Tier: {all NPIs, all negative elements, *the*}
- Constraints: { $X \cdot \text{the}$, $X \cdot \times$ }, where X is an NPI

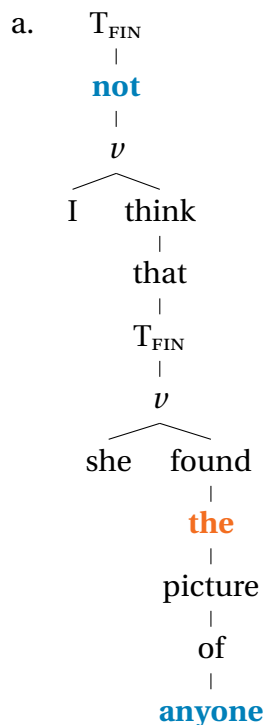
The dependency trees for the preceding examples are shown below, along with the c-strings and tiers for the NPIs.

(30) I didn't think that she found a picture of anyone.



- b. C-string: **anyone** · of · picture · a · found · she · ν · that · think · I · ν · **not** · T_{FIN}
 c. Tier: $\times$ · **anyone** · **not** · $\times$

(31) * I didn't think that she found **the** picture of anyone.



- b. C-string: **anyone** · of · picture · a · found · she · ν · that · think · I · ν · **not** · T_{FIN}
 c. Tier: $\times$ · **anyone** · **the** · **not** · $\times$

Note that this analysis already allows multiple NPIs to appear in the same command string as long as they are adjacent, and that the chain of NPIs is eventually ended by a licenser. As such, we can also handle data like the following.

(32) Nobody gave anything to anyone for any reason.

While there is much more to be said about NPI licensing conditions, they are beyond the scope of this dissertation. For example, subjects are d-commanded by sentential negation in their base position, yet are not licensed, as in **Anyone didn't see her*. Conversely, NPIs *can* be licensed by a moved negative element, as *Nobody ever seems to want to take responsibility*. Proper treatment of NPI-licensing with c-strings therefore requires the ability to track derived positions of movers, among other enhancements. Since we will ultimately use c-strings here only for phenomena which obey the d-command generalization as calculated using the base positions of all elements, these complications will not trouble us further. Instead, we focus our attention on another command-based constraint which is highly relevant to case and agreement: relativized minimality.

Sidebar 2.6: What about binding?

The analysis of NPI licensing presented above differs markedly from that presented by Graf and Shafiei (2019), who use SS-TSL rather than TSL. The motivations for their analysis are two-fold. First, certain binding constraints, particularly Principle B, are not TSL under common assumptions; indeed, Graf and Shafiei are forced to simplify the problem somewhat just to obtain a SS-TSL analysis of Principle B. Second, by using a more powerful tier projection function, they are able to obtain a set of tier-based constraints which are not just strictly local, but *finite*. This means that a single tier can be constructed which contains all constraints for all licensing dependencies (TSL languages can be intersected in full generality if their tier alphabets are either identical or fully disjoint, but this may not be possible if the tier alphabets are only partly overlapping.) Since we are already assuming the existence of multiple tiers, and do not aim to capture binding constraints at all, these considerations do not apply here.

An alternative approach by Ikawa and Jardine (2021), also using c-command strings, treats binding as kind of transformation, using string-to-string maps rather than constraints. They note that from this perspective, binding looks rather similar to long-distance dissimilation in phonology. Although I do not use maps in this work, it is possible that this perspective could also shed light on certain aspects of case and agreement. This topic will be revisited in Chapter 3 and Chapter 6.

2.7.3 Relativized minimality

For our purposes, c-strings are crucial in that they allow us to enforce RELATIVIZED MINIMALITY (Rizzi 1990, 2001, 2013), which is generally not possible with tree tiers.

The principle of relativized minimality holds that a syntactic operation (movement, agreement, etc.) involving two elements X and Y cannot hold if there is another element Z which intervenes and which could also participate in said operation. In other words, the closest potential mover to some landing site is the one that must actual move, and the closest agreement target for an agreeing head must be the one that agrees. By hypothesis, the ordering relation which defines intervention is c-command, or in our case, d-command.

(33) **Relativized minimality (abstract principle):**

An operation O that targets X and Y can only apply if there is no Z s.t. X d-commands Z and Z d-commands Y and Z can undergo O .

This principle can be operationalized in various ways in different syntactic frameworks. In Minimalism, it is naturally stated in terms of syntactic features, possibly making reference to whether they have already been valued/checked. In the MG dependency tree, it should be stated in terms of potential and actual bearers of movement/agreement features.

(34) **Relativized minimality (MG dependency trees)**

If $X_{[+F]}$ d-commands $Y_{[-F]}$, then there can be no Z such that X d-commands Z and Z d-commands Y and Z is able to bear $-F$.

We must also make explicit the relation between variants of a lexical item with and without some licensee feature. In the case of *wh*-movement, we include all items in $\{\text{who}_{[-\text{Wh}]}, \text{what}_{[-\text{Wh}]}, \text{where}_{[-\text{Wh}]}, \dots\}$ as well as their counterparts in $\{\text{who}, \text{what}, \text{where}, \dots\}$. The latter set is calculated as follows, where Lex stands for the MG lexicon.

Definition 10 (potential bearers of $-F$)

$$\text{can-bear}(-F) \stackrel{\text{def}}{=} \{l \in \text{Lex} : l_{[-F]} \in \text{Lex}\}$$

Now, no *wh*-item may intervene between a *wh*-mover and its landing site, whether or not intervener does in fact bear $-\text{Wh}$. If it does bear the feature, then we violate the rule that a given landing site can only have one mover. This is already handled by our tree grammar from Section 2.6.2. If it does not, then relativized minimality is violated.

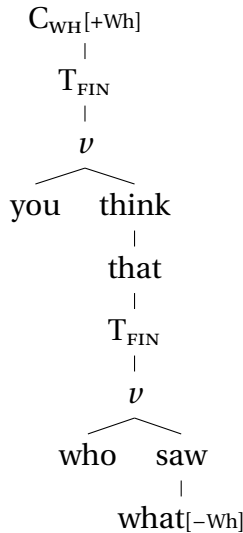
To implement the minimality constraint over c-strings, we need only to project all *wh*-movers and landing sites (potential or actual), and disallow an item bearing $-\text{Wh}$ from being immediately followed by another *wh*-element. But this is just one instance of a more general constraint, which is that no mover can be immediately followed by *anything* other than a landing site. We therefore use the more general version.

This grammar is shown in (35) below. The first constraint $(*X_{[-\text{Wh}]} \cdot Y)$ represents relativized minimality, while the others represent (a subcase of) multiple *wh*-movement and a mover with no landing site, respectively. An example of a minimality violation for *wh*-movement is given in (36), with c-strings in (37). With a few additional tweaks, this forms the basis of the approach to agreement outlined in Chapter 6.

(35) C-string grammar for minimality in *wh*-movement

- Tier: $\{\text{all } wh\text{-movers, all } wh\text{-landing sites}\}$
- Constraints: $\{X_{[-\text{Wh}]} \cdot Y, X_{[-\text{Wh}]} \cdot Y_{[-\text{Wh}]}, X_{[-\text{Wh}]} \cdot \times\}$ for all X, Y

(36) Tree for *What do you think that who saw?*



(37) C-strings for ‘you’, ‘who’, and ‘what’

you	c-string →	$\text{you} \cdot \nu \cdot T_{\text{FIN}} \cdot \mathbf{C_{WH[+Wh]}}$
	project →	$\times \cdot \mathbf{C_{WH[+Wh]}} \cdot \times$ (no violations)
who	c-string →	$\mathbf{who} \cdot \nu \cdot T_{\text{FIN}} \cdot \text{that} \cdot \text{think} \cdot \nu \cdot \text{you} \cdot T_{\text{FIN}} \cdot \mathbf{C_{WH[+Wh]}}$
	project →	$\times \cdot \mathbf{who} \cdot \mathbf{C_{WH[+Wh]}} \cdot \times$ (no violations)
what	c-string →	$\mathbf{what} \cdot \text{saw} \cdot \mathbf{who} \cdot \nu \cdot T_{\text{FIN}} \cdot \text{that} \cdot \text{think} \cdot \nu \cdot \text{you} \cdot T_{\text{FIN}} \cdot \mathbf{C_{WH[+Wh]}}$
	project →	$\times \cdot \mathbf{what} \cdot \mathbf{who} \cdot \mathbf{C_{WH[+Wh]}} \cdot \times$ (mover followed by mover)

We now find ourselves in an interesting situation. On the one hand, certain constraints on movement can only be handled with tree tiers or tiers over c-strings, while others can be handled with both to some extent. Our c-string constraint $*X_{[-Wh]} \cdot Y_{[-Wh]}$ cannot handle all cases of multiple movers (specifically, those in which the movers do not themselves stand in a d-command relation), but the constraint $*X_{[-Wh]} \cdot \times$ does indeed rule out all cases of a mover without a landing site. Should such constraints be encoded in both the tree tier grammar and the c-string grammar? If not, then where should they be placed?

One speculative answer is that upward-looking constraints should be modeled with c-strings, and downward-looking constraints with tree tiers. This is due to the asymmetric geometry of a tree: searching downward involves making a choice of which branch to follow, while upward search presents no such choice. This idea is implicit in the automaton model of Graf and De Santo (2019), which is discussed in Chapter 6 as part of the motivation for the proposed model of agreement. Thinking more broadly, it really does not matter which way we choose, as each perspective can potentially yield interesting insights. It may also be possible to unify these two perspectives; see Graf and Hanson (2025) for a first attempt. For the purposes of this dissertation, what matters most is that we know the limits of each approach. We will therefore do much as we can within a single type of tier-based grammar while acknowledging that certain constraints ultimately require the other type.

2.8 Caveats and limitations

To close this chapter, I wish to mention a few outstanding formal issues. First, some readers may be concerned that TSL grammars, for all of their merits, are still too powerful, as it is fairly easy to construct examples which seem not to correspond to any realistic linguistic pattern. This is correct, and as far as I can tell, unavoidable. It is never possible to rule out every unattested pattern on purely formal grounds. We also need to make use of constraints of language acquisition, of diachronic development, and so on. There may also be other formal constraints which are quite separate from subregular complexity. Graf (2019, 2020b) discusses monotonicity as an important constraint in multiple modules of grammar. As it stands, TSL over trees marks a major improvement over previous results in this area.

Second, there is an insidious loophole which affects not only the present model, but essentially all models of syntax, and which subverts all subregular complexity results. This is called “feature coding” by Graf (2020a), and boils down to construction of ad hoc syntactic categories which implicitly encode long-distance dependencies. This trick allows an SL grammar (or any kind of subcategorization mechanism) to simulate any regular grammar, for both string and tree languages. This connects to the issue of abstraction in wider linguistic theory, namely what kinds of abstract categories are legitimate to posit. In this case, my view is that only an acquisition theory can provide a proper answer. In the meantime, all we can do is to avoid positing abstract categories without strong independent evidence.

2.9 Summary

In this chapter, I provided a crash course on subregular string and tree languages and their application to linguistics. Of the various subregular classes, TSL-2 encapsulates a model of relativized locality which is appropriate not just for phonology but also syntax. The idea of strictly local constraints on a tier can be adapted to syntax in several ways, which we categorized into tree tiers and tiers over paths. We saw how TSL-2 tree grammars allow us to model the constraints on long-distance syntactic dependencies such as movement and NPI licensing, and showed how locality constraints such as islands and relativized minimality can be understood in terms of blockers on a tier. We also discussed how small differences in the TSL-2 grammar correspond to actual linguistic variation, and conversely, how many types of unattested patterns cannot be expressed at all. I build on these ideas in the remainder of this dissertation in order to show that TSL-2 is also an appropriate model for the syntax of case and agreement.

Chapter 3

Modeling Case

3.1 Overview

In this chapter, I adapt the tree tier of system of Graf and Kostyszyn (2021) to the syntax of case, forming the basis of the studies in the next two chapters. This sort of analysis was first performed by Vu et al. (2019) using SS-TSL over traditional MG derivation trees. Here, I use (pure) TSL over MG dependency trees, mirroring Graf’s (2022) analysis of movement. In a nutshell, we introduce additional MG features indicating the cases of nominal heads. Our job, then, is to show that the distribution of these features can be modeled concisely with one or more TSL grammars.

The general approach is highly reminiscent of the Case in Tiers framework (Yip et al. 1987) as well as Dependent Case Theory (Marantz 2000; Baker 2015). Although I do not provide a direct comparison with these theories, readers who are familiar with them should be able to see how the kinds of case patterns which motivated them are highly natural when considered in terms of tree tiers.

The basic idea is that all case assignees (roughly, all DPs) are brought together as daughters of a case assigner (such as $C/T/v$) on a tier, allowing their cases to be determined in a local manner. As a preview, a simple English example is illustrated in Figure 3.1, showing the dependency tree and the proposed case tier for the simple ditransitive sentence *They showed us themselves*. Here, finite C is assumed to be the case assigner for all DPs in the clause. The relevant tier therefore contains at least these two categories, perhaps among others. For now, all other nodes are ignored, leaving the three D heads as daughters of C .

By hypothesis, the first DP should be nominative and any others accusative, corresponding to the generalization that, in English, it is the highest DP in the clause which receives nominative case. The constraints on the daughter string language of C therefore look something like the following:

$$(1) \quad G^+ = \{ \times \cdot D_{[NOM]}, D_{[NOM]} \cdot D_{[ACC]}, D_{[ACC]} \cdot D_{[ACC]}, D_{[NOM]} \cdot \times, D_{[ACC]} \cdot \times, \times \cdot \times \}$$

This chapter is based on material from Hanson (2025b) “The Typology of TSL Case Assignment”. It greatly expands on the background sections of that paper, particularly the English case analysis. Sections 3.2 and 3.4 are completely new.

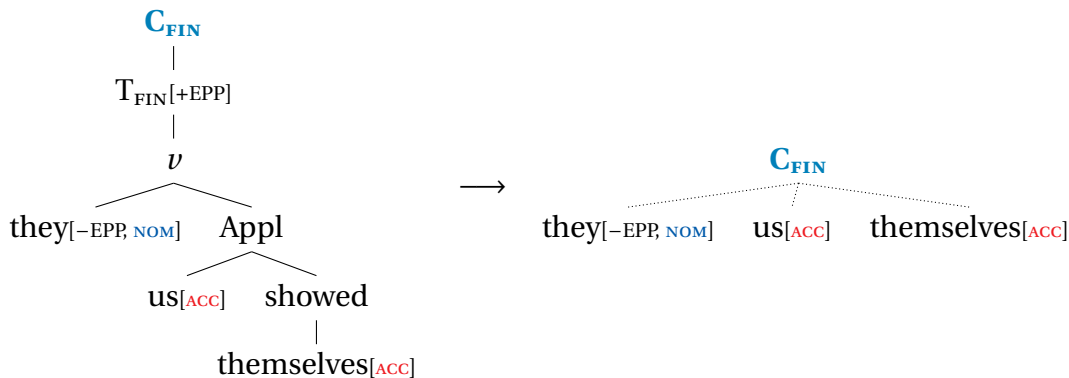


Figure 3.1: MG dependency tree and case tier for *They showed us themselves*.

Later in this chapter, this analysis will be fleshed out to cover a variety of common English structures, including embedded clauses, the internal structure of DPs, gerunds, and PPs. Importantly, the particular analysis is not meant to be comprehensive or incontrovertible, rather, it has been chosen to be representative of the approach that will be taken in Chapters 4 and 5. Indeed, the fact that English is so poor in case marking means that there are numerous analyses which are compatible with the basic data. For our purposes, what is important is that we can achieve an analysis which is simple and direct, covers many important structures discussed in the literature, and requires only a single tier.

But first, there are a few preliminaries that need to be dealt with, such as:

- What counts as case in the syntactic sense?
- How do we model case in the dependency tree?
- What are the core facts to be accounted for?
- What issues can we safely ignore?

I therefore begin with an overview of the types of data falling within the scope of investigation (Sections 3.2 and 3.3), followed by a discussion of some theoretical issues and how we can abstract away from them here (Section 3.4). I then motivate and present the analysis of case in English, which I again emphasize is purely intended to illustrate the general approach (Section 3.5), followed by a summary of some of the formal issues that arise (Section 3.6). The chapter finishes with a discussion of some big picture questions (Section 3.7). This sets the stage for the typological study in Chapter 4 and the analysis of Japanese in Chapter 5.

3.2 What is case?

Case is a multi-faceted phenomenon. It is both morphological and syntactic, and is relevant at both the level of the noun and the entire nominal phrase, whether the latter is conceptualized as a NP/DP/K(ase)P/etc. (I use the term *nominal phrase* or *nominal* here in order to emphasize that the category of the phrase is not at issue.)

As a starting point, it is a formal property of nominals realized in their overt form (morphology) which reflects the context they appear (syntax). In English, we can distinguish

perhaps three formal categories, which we call nominative, accusative, and genitive case. (I leave aside the *mine/yours/his/hers* series, whose distribution is extremely restricted.) In their canonical function, these correspond to the subject of a finite clause, objects, and possessors, respectively.

$$(2) \quad \left\{ \begin{array}{l} \text{They} \\ * \text{Them} \\ * \text{Their} \end{array} \right\} \text{ sent } \left\{ \begin{array}{l} \text{me} \\ * \text{I} \\ * \text{my} \end{array} \right\} \text{ to examine } \left\{ \begin{array}{l} \text{his} \\ * \text{I} \\ * \text{him} \end{array} \right\} \text{ handiwork.}$$

English is relatively impoverished in that ordinary nominal phrases are only marked for genitive case, realized with a phrase level clitic, e.g. [*the king of France*]'s favorite wine. The full set of distinctions is realized only in the personal pronouns, where case is fused with person/number/gender with a small degree of syncretism. The standard paradigm is shown below.

(3)

Pronoun	Nominative	Accusative	Genitive
1SG	I	me	my
1PL	we	us	our
2.SG/PL	you	you	your
3SG.M	he	him	his
3SG.F	she	her	her
3SG.N	it	it	its
3PL	they	them	their

Many other languages have much richer case systems, with more case distinctions and considerable variation in both the morphological and syntactic details. In Japanese, every case has a unique clitic, consistent across all nominal phrases (see Chapter 5). In many languages, the noun itself and other major constituents such as adjectives and demonstratives are marked with the case of the entire nominal phrase. Additionally, case endings may be fused with person/number/gender, and different classes of nouns may have different sets of endings; these are known as declension classes. Latin and the modern Slavic languages exemplify such a system. Since most languages with case also have adpositions, and there is some overlap in their function, and it is not always clear whether a given morpheme is a case marker or an adposition. Thus, the claim that languages like Finnish and Hungarian have dozens of cases depends in part on where we draw this line. Some languages combine aspects of both systems: every case-marked noun phrase gets a clitic/adposition, which in turn controls the inflection of the head noun; Hindi is one such language. These and other intricacies of morphological realization, though extremely interesting, will be set aside, as they are almost completely orthogonal to the primary syntactic question: given a nominal phrase in syntactic context X, what can/must its case be?

Though seemingly simple, there are subtleties on the syntax side as well. For example, how exactly should we encode case distinctions in syntax? It is conventional to treat case as a FEATURE or set of features, analogous to other morphological features such as tense, aspect, person/number/gender, and so on. This is especially natural when case is realized as an

inflection, as is common in the Indo-European family. However, it is somewhat awkward in languages like Japanese where case markers look more like semi-independent words, and would be more naturally modeled with a separate K(ase) head. Fortunately, there is not much difference from a formal perspective, as we can easily realize a morphological feature as an adjacent word or fuse a K head with noun as part of the mapping from syntax to morphology. This is not to say that there is no difference whatsoever between these two types of theories, only that they close enough that the differences do not much affect how we approach the question of external distribution. As a matter of practice, I will combine the two systems for Japanese, treating each case marker as a K head with a unique case feature, allowing the same general format to be used in the TSL grammars regardless of language.

A second complication has to do with the external and internal distribution of case in the nominal phrase. To determine the case of the entire nominal, we only need a feature on the head of the highest extended projection, whether it be N/D/K/etc. Let us call the process which determines this feature *CASE ASSIGNMENT*. Case assignment is assumed to be determined contextually by various aspects of the syntactic structure, and is also influenced by lexical and semantic factors, as discussed below. However, we must also specify how case is duplicated on various heads and constituents of nominal phrases, what is often termed *CASE CONCORD*. For reasons that will be discussed in Chapter 4, this is perhaps better seen as a kind of agreement, and will be treated briefly in Chapter 7.

To briefly summarize, we have case features on various heads within the nominal phrase which correspond more-or-less directly to morphological cases. Our primary concern is determining the case of the head of the entire nominal phrase, which we call case assignment, as opposed to concord within the nominal phrase, which we set aside. (Note: for expository convenience, I will assume nominal phrases to be DPs for the remainder of this chapter.) This settled, let us now tackle some theoretical and methodological issues regarding the syntax of case. For further general background on case, the reader is referred to Blake (2001), Butt (2006), and Malchukov and Spencer (2008).

3.3 Modes of case assignment

On the syntactic side, cases can be organized into three rough categories based on the specific kind of contextual information that determines their distribution: *STRUCTURAL* case, *LEXICAL* case, and *SEMANTIC* case. Importantly, these categories are not mutually exclusive: a single case might have several modes of assignment. For present purposes, these will be treated as distinct cases with the same surface forms, e.g. *LDAT* for lexical dative.¹

STRUCTURAL CASE includes most instances of nominative, accusative, ergative, and genitive case, and some instances of dative. Such cases appear in regular recurring contexts which are not specific to individual lexical predicates. For example, the subject of a finite clause may receive nominative case while the subject of a non-finite clause receives accusative case.

¹A common alternative is to assume a categorial difference, whereby the structural case-marked version is a DP and the corresponding lexical case involves a covert PP; this would also work here. As with the above issue of case features vs. case heads, these alternatives are interchangeable for our purposes.

Structural cases are by far the most interesting in terms of syntactic distribution, since the elements which are thought to influence them may occur at a distance, necessitating the use of one or more non-trivial tiers. They will therefore be our focus.

In contrast, LEXICAL CASE is assigned to arguments of specific lexical predicates. For example, a given verb might specify that either its subject or object is dative, regardless of the case that would be ordinarily assigned in that position. Generally speaking, the set of predicates which follow a given lexical case rule is highly idiosyncratic, though there are some recurring tendencies. In any case, since arguments are assumed to be local to their selecting heads, lexical case is somewhat uninteresting from a formal perspective. But we cannot ignore it entirely, since it interacts with structural case in an interesting way, namely that structural case should be assigned iff lexical case is not. Also, it most likely occupies a different tier (the tier which contains everything); see Section 3.7.3 for details.

Finally, we have SEMANTIC CASES such as locative and instrumental, which occur mainly on DP adjuncts. These are essentially self-licensed, and are therefore of little interest to us. In terms of distribution, they can be assumed to be wrapped in a covert PP; this also applies to the corner case in which semantic case-marked nominals appear in argument position. Sections 3.5 and 3.6 provide some techniques for dealing with PPs more generally.

Note that some theories distinguish an additional category of INHERENT CASE, which is assigned to an argument DP together with a specific thematic role. This is claimed to include at least some instances of ergative case. (I deal with ergative case to some extent in Chapter 4.) For our purposes, it can be seen as a subtype of lexical case which obeys a productive semantic generalization.

3.4 Some theoretical (non-)issues

In this final background section, I briefly address some issues which might be on the minds of syntacticians, but which we can safely abstract away from for present purposes.

3.4.1 Assignment vs licensing

In this thesis, I primarily use the language of case ASSIGNMENT: head X is assigned case Y in context Z. However, the analysis is completely compatible with an interpretation in which case features enter the derivation on the relevant nominal head, and are LICENSED after the fact: any derivation that violates the case licensing rules is deemed illicit. In one sense it is hard to distinguish these two interpretations, as the presence of a case feature on a given node in the MG dependency tree means that said node will be assigned said case at some point during the derivation.

From a purely computational perspective, the present approach is a pure licensing account: the TSL-2 tree grammar says whether or not a given configuration is well-formed, period. In this sense, the natural alternative is to have no case features in the syntax, but rather introduce them as part of the mapping from the dependency tree to a derived structure. Case and agreement have not yet been studied in terms of subregular maps/transductions,

though movement has (Graf 2023). Since subregular languages differ somewhat in their properties from the corresponding subregular maps, it would be useful to investigate how case and agreement might work in the mapping.

3.4.2 The timing of case assignment

Related to the above, I make no strong claim about whether case (and also agreement) is assigned within syntax proper as opposed to the mapping to morphology (“spell-out”). This is relevant since case is often assumed to be assigned late (cf. Bobaljik 2008), even though this need not be so (cf. Preminger 2014). By my judgment, it is very hard to make a well-grounded claim either way. Empirically speaking, the key issue seems to be whether there is a principled ordering between these operations, as well as movement. However, interactions between movement, case, and agreement can often be interpreted in several ways, such that it is unclear which is the causal factor. (Chapter 4 addresses this issue with regards to case and movement.) In computational terms, the distinction between subregular languages and maps just mentioned is also relevant. For now, I treat movement, case, and agreement uniformly using constraints on the derivation tree, leaving open the possibility that some other model might be superior for some operations.

Sidebar 3.1: What is post-syntax anyway?

The above discussion takes for granted that the derivation tree corresponds to syntax, and the mapping to post-syntax. This is the most natural interpretation, but there also some ambiguity. After all, movement is only really enacted in the mapping; the same is true of agreement in either the system presented in Chapter 6 or in the “feature channel” system of Ermolaeva (2018) and Ermolaeva and Kobele (2022). Thus, the present model could also be interpreted as saying that movement and agreement are both post-syntactic, whereas case comes closest to looking like a bona fide syntactic operation. Furthermore, if we split up the mapping into several steps—first enact movement and agreement, then handle linearization and other aspects of morphology—we end up with a system resembling the classic Aspects model (Chomsky 1965), in which the first mapping (from deep structure to surface structure) is firmly part of syntax.

3.4.3 How many case features?

In the preceding section, I was ambiguous about whether there is a single case feature which takes on multiple values, as in Chomsky (2000) and much following work, or whether there is a set of separate (privative) features, one for each case. For our purposes, this makes no difference whatsoever. $D_{[\text{case}:\text{NOM}]}$ and $D_{[\text{NOM}]}$ are trivially interchangeable, with the proviso that no lexical item is allowed to bear more than one case feature. (Relaxing this restriction would be one way to model case stacking, the phenomenon in which a single nominal bears multiple overt case markers, though I do not attempt this here.) I therefore take the privative option in order to minimize notational clutter in the TSL grammars.

An additional consideration relevant to the MG-based framework utilized here is whether to use binary features, e.g. $\pm\text{NOM}$ to indicate assigners/assignees of nominative case, mirroring the those for selection and movement. This has precedent in the MG literature (cf. Laszakovits 2018). However, I have decided to use unary features which appear only on case assignees, with no overt marking on case assigners. Again, this is mainly for conciseness, but it might be further justified given that there is no reflex in the morphology (or semantics) for case assigners, only for case assignees.²

3.4.4 Morphological vs. abstract case

As discussed above, this work is primarily concerned with the syntactic distribution of case, not the details of morphological realization. It is also not concerned with constraining the distribution of overt/covert nominals, as in (Vergnaud 1977; Chomsky 1981). This more modest version of (syntactic) case has been termed MORPHOLOGICAL CASE or m-case by Baker (2015) and others in order to distinguish it from what Chomsky (1981) calls ABSTRACT CASE or ‘Case’, which in addition to informing the morphology, also determines whether or not a DP may appear overtly via a constraint known as the CASE FILTER. (In the classic version, case-marked DPs must be overt, and caseless DPs must be realized as PRO.)

This terminology is unfortunate, since m-case is very much syntactic, and also extremely abstract. For example, it allows for cases to undergo complete neutralization on the surface, as in Legate’s (2008) analysis of Warlpiri. That said, we would not posit case in a language like Chinese, which has no morphological realization in any context; alternatively, such languages have a single case corresponding to a zero affix.

I ignore the case filter here because I consider it to be orthogonal to present concerns. It would be perfectly possible to construct a TSL grammar which fails to assign case to DPs in certain positions, and add to this a surface filter which allows only PRO in such positions. As for whether this is advisable, McFadden (2004) provides extensive empirical arguments against the supposed anti-correlation between case and overtness, even for the English data which first motivated it.

3.4.5 Is case distinct from agreement?

In recent Minimalist theory, case is often assumed to be assigned by the same Agree operation which underlies agreement, starting with the original work of Chomsky (2000). A notable exception is Dependent Case Theory (Marantz 2000; Baker 2015), whose advocates distinguish dependent case assignment as an independent operation, either in addition to the possibility of case assignment under Agree (Baker and Vinokurova 2010) or as the only mode of case assignment (Levin and Preminger 2015). My own position is that case and

²Another reason not to pursue a binary feature system is that there does not seem to be anything close to a one-to-one relationship between case assigners and assignees. While the relationship between movers and landing sites can be many to one (recall the example of multiple *wh*-movement from Chapter 2), case strains this relationship much further: one case assigner can be responsible for multiple distinct cases, and default cases can be thought of as having no assigner at all.

agreement probably should be unified, if only at an abstract computational level: it is trivially true that case can be reinterpreted as a kind of agreement, for both case and agreement are of the same computational complexity. But then, so is movement, and perhaps other operations that are worth distinguishing on conceptual grounds.

As such, I consider recent arguments about whether case can/should be folded into Agree (Preminger 2024; Bárány and Sheehan 2024; Poole 2024; Clem and Deal 2024) to be largely beside the point. If the Agree algorithm is formulated in a way which is sufficiently general to handle all agreement patterns, it is all but guaranteed to work for case as well. As best I can tell, the waters have been muddied by conflating formal questions about the distribution of case (does case sometimes depend on the presence of nearby DPs?) with substantive questions (does it always co-occur with agreement, or can it apply independently?). This is compounded by the addition of various auxiliary assumptions which have been added to the framework of Dependent Case Theory over the years, whose empirical viability is independent of the existence of dependent case as an empirical phenomenon. For present purposes, I will treat dependent case as just one case phenomenon among many starting in Chapter 4.

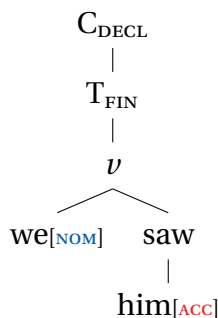
3.5 TSL case assignment – an example

In this section, I demonstrate the TSL-2 model of case by constructing a partial grammar for English, covering a variety of common structures. Recall from the introduction that all case assignees (approximately, all DPs) will be brought together as daughters of a case assigner on a tier, allowing their cases to be determined locally. The basic approach mirrors that taken in Chapter 5 for Japanese, but on a smaller scale. In particular, we will employ only a single tree tier. We start by identifying basic descriptive generalizations, then encode them in a TSL-2 tree grammar, which is subsequently refined to handle additional corner cases. For reasons of brevity, the final step is truncated, with most of the refinements sketched informally but not fully implemented.

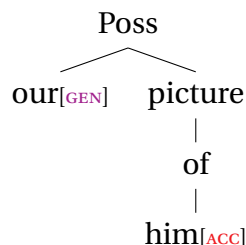
3.5.1 Setup and descriptive generalizations

We begin with the dependency tree representation. We add a new category of CASE FEATURES to our dependency trees. As discussed above, these are unary, appearing only on DPs and other categories which receive case. For English, we need only three: **NOM**, **ACC**, and **GEN**, which we assume to occur on all D heads even though only pronouns show the full paradigm. A pair of examples illustrating all three cases is given below. For the fragment DP, the root is left caseless for now, though as will be discussed shortly, it should probably be accusative.

- (4) a. [CP We saw him]



- b. [DP your picture of him]



Now, we consider what elements must appear on the tier. In principle, these should include all dependent elements, that is, all case assigners and assignees, perhaps accompanied by miscellaneous blockers. This is analogous to the analysis of *wh*-movement from Section 2.6, which included all *wh*-movers and landing sites as well as island nodes. Certainly, all D heads must certainly be included, for otherwise we cannot control their cases at all (we discuss below whether other categories might also bear case). But what about the case assigners? These are not trivial to determine since, as mentioned earlier, there is no direct evidence as to which nodes are responsible.

The general solution I advocate (explored in much greater detail in Chapter 5) is to start by identifying broad descriptive generalizations for where each case may appear, providing an initial hypothesis for the domain nodes, including both assigners and blockers. In English, these are fairly clear. First, there is a parallelism between nominative and genitive: each occurs in the structurally highest position in the finite clause and DP, respectively, and nowhere else. Nominative is never found in any position other than the canonical subject position (leaving aside obvious cases of prescriptivism such as *It is I!*), even if this position is filled by a CP or PP, nor can non-nominative DPs ever appear in the canonical subject position.

- (5) Subject of finite clause must be nominative

- a. {I/*me/*my} {slept/found it/gave it to them}.

- (6) Non-subjects cannot be nominative

- a. I found {them/*they}!
 b. [That she laughed] enraged {them/*they}. (CP subject)
 c. [Under the bed] was {?him/*he}. (PP subject)

- (7) Non-finite subjects cannot be nominative

- a. {us/*we} winning the war (acc-ing gerund)
 b. for {us/*we} to win the war (for-to infinitive)

The same can be said of the internal constituents of DPs. The nominal possessor or subject must be genitive, and other DPs must not be.³

³Again, we leave aside the mine/yours/his/hers series, which have a possessive meaning but are clearly distinguishable in form and position from normal genitives.

- (8) DP subject/possessor must be genitive
- {our/*us/*we} picture of {him/*he/*his} (possessor)
 - {his/*him/*he} admiration of {them/*they/*their} (thematic subject)

A small complication occurs with gerunds, which are the major type of nominal clause in English, and come in several flavors. The inner layer may be an NP or VP, and the outer layer a DP or TP accordingly; in addition, it is possible to embed a VP in a TP (cf. Abney 1987). In all of these structures, the generalization is upheld.

- (9) Three types of gerunds
- [_{DP} their [_{NP} hugging of us]] (nominal gerund)
 - [_{TP} them [_{VP} hugging us]] (verbal gerund)
 - [_{DP} their [_{VP} hugging us]] (hybrid gerund)

Accusative, then, is the default case, as evidenced by its wide distribution (object of VP, object of PP, subject of non-finite TP) in combination with the fact that fragments are accusative (Who did it? Him!). Putting all of this together, we have a set of three descriptive generalizations.

- (10) Descriptive generalizations for case in English
- A DP subject of a finite clause is nominative
 - A DP subject/possessor of a DP is genitive
 - All other DPs are accusative

3.5.2 Constructing the grammar

We now translate these generalizations into a TSL-2 tree grammar, starting with the tier alphabet. We already decided that all D heads must be included. Next, either finite C or T needs to be included so that it can assign nominative case to the subject. Let us choose finite C, since this allows for a slightly more elegant analysis: we simply project *all* C heads, including non-finite *for*, which is often taken to be a case assigner in other theories. Genitive case is assigned by D, so no further addition is needed. Finally, we said that accusative is the default case, so it does not have a dedicated assigner, though we may include additional domain nodes (blockers) in order to prevent DPs from receiving nominative or genitive case when they should not. Prepositions are a good candidate for such a blocking category; we will see several examples below. Our tier alphabet is therefore as follows:

- (11) Tier alphabet for English case tier
- all D heads
 - all C heads
 - all P heads

As a general principle, we should avoid choosing an excessively ad hoc set of elements for the tier alphabet, though as we saw in Chapter 2, some arbitrary variation is necessary, and in fact predicted. However, without a detailed acquisition theory, we cannot say much more than this. As a matter of practice, I follow the linguist’s intuition that we should be dealing in NATURAL CLASSES of elements as much as possible, if only because this reduces the degrees of freedom available to the analyst, hence the decision here to take as our tier alphabet all elements of categories D/C/P.

Next come the constraints, which can also be derived from our descriptive generalizations in a straightforward manner. In this case, we can approximate the notion of “highest DP in a domain” as “first daughter of a domain node on the tier”.⁴

(12) Semi-formal constraints for English case tier

1. NOMINATIVE CONSTRAINT: If the first tier child of a finite C head is a D head, then it must be nominative. No other D may be nominative.
2. GENITIVE CONSTRAINT: If the first tier child of a D head is a D head, then it must be genitive. No other D may be genitive.
3. ACCUSATIVE CONSTRAINT: If a D head is not the first tier child of C or D, then it must be accusative.

This has some limitations, as it is possible for a node X to be the left sister of Y on the tree tier even though X does not c-command Y in the full structure. We will address this shortly. In the meantime, we translate the constraints into SL-2 daughter string languages; later we will extend them to TSL-2. The daughter string language for C_{FIN} , for example, is as follows:

(13) Daughter string grammar for C_{FIN}

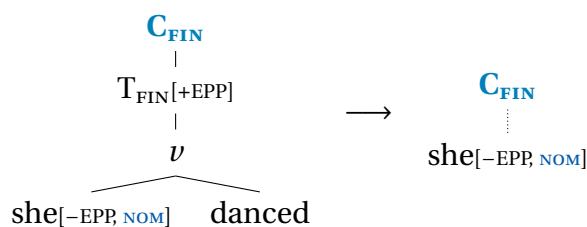
$$G^+ = \{ \times \cdot D_{[\text{NOM}]}, D_{[\text{NOM}]} \cdot D_{[\text{ACC}]}, D_{[\text{ACC}]} \cdot D_{[\text{ACC}]}, D_{[\text{NOM}]} \cdot \times, D_{[\text{ACC}]} \cdot \times, \times \cdot \times \}$$

This grammar allows daughter strings such as: $D_{[\text{NOM}]}$, $D_{[\text{NOM}]} \cdot D_{[\text{ACC}]}$, $D_{[\text{NOM}]} \cdot D_{[\text{ACC}]} \cdot D_{[\text{ACC}]}$, etc., corresponding to intransitive, transitive, and ditransitive structures, respectively. At the same time, it disallow an accusative subject followed by a nominative object, among many other illicit configurations. It also allows an empty daughter string, allowing weather predicates to be treated as having no arguments whatsoever; if this is not desired, the final bigram $\times \cdot \times$ can be removed.

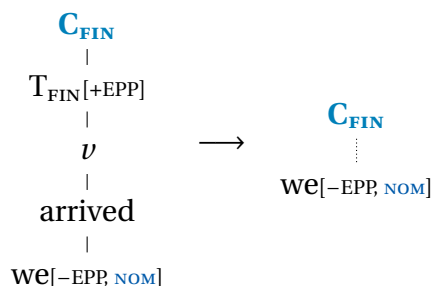
Let us confirm that this works as intended. The simple transitive structure from (4a) is repeated below with its corresponding tree tier, as is the ditransitive example from Figure 3.1. For completeness, two intransitive structures are included as well: one with an external argument (an unergative verb) and one with an internal argument (an unaccusative verb). Domain nodes are bolded, and the top/bottom edge markers are omitted for simplicity. While I have not included any examples of illicit case configurations in the same structures, it should be clear how they would be ruled out.

⁴This use of order among tier sisters is novel to the present work. See Section 3.7.1 for discussion.

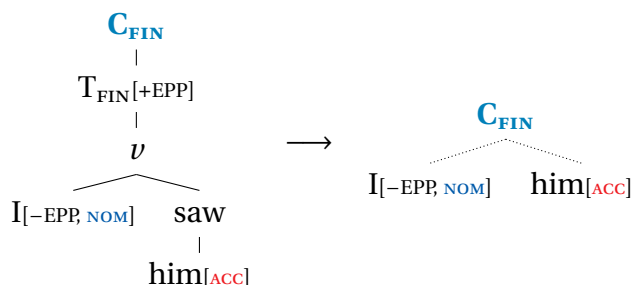
- (14) She danced. (unergative)



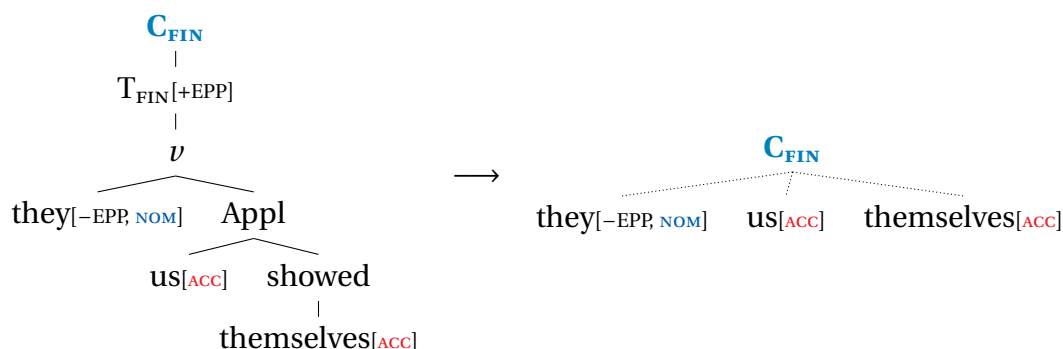
- (15) We arrived. (unaccusative)



- (16) I saw him. (transitive)

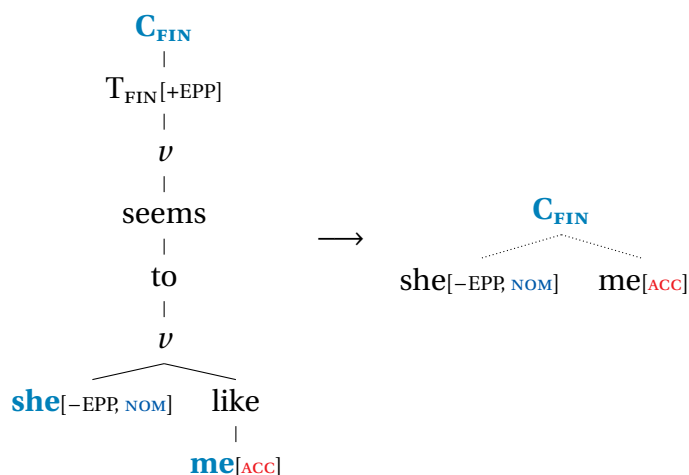


- (17) They showed us themselves. (ditransitive)

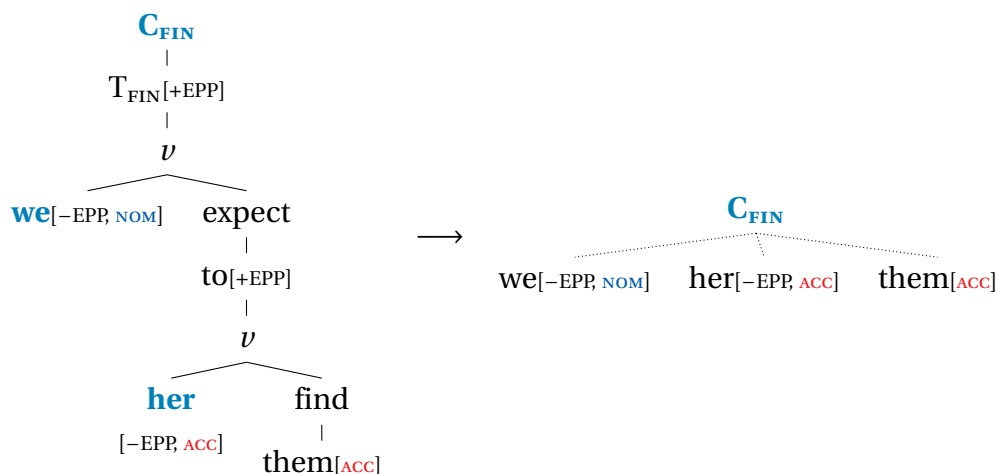


Since infinitive *to* does not appear on the tier, the analysis immediately extends to sentences containing embedded infinitives, including raising infinitives as well as ECM infinitives. These are illustrated below. Notice that nominative tends to co-occur with $-EPP$, since this highest DP in the clause is also the one that must move to Spec-TP. But the correlation is not perfect, since certain infinitives, including ECM as well as *for-to* infinitives (discussed below), have an overt subject in clause-initial position which remains accusative. We revisit the role of the $-EPP$ feature in Section 3.6.3.

(18) She seems to like me (raising)



(19) We expect her to find them (ECM)



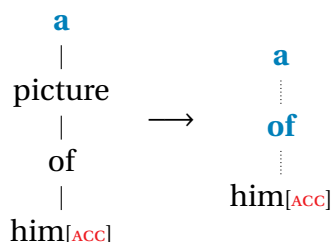
Moving on to the internal structure of DPs, our second daughter string language is as follows.

(20) Daughter string grammar for D

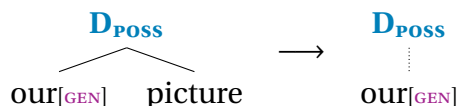
$$G^+ = \{ \times \cdot D_{[GEN]}, D_{[GEN]} \cdot D_{[ACC]}, D_{[ACC]} \cdot D_{[ACC]}, D_{[GEN]} \cdot \times, D_{[ACC]} \cdot \times, \times \cdot \times \}$$

For normal DPs, this trivially does what we want, for there can never be more than one argument which is not embedded in a PP: the possessor. This is because nouns obligatorily select PP complements in English. The basic paradigm, building on the example in (4b), is shown below. For simplicity, I assume that the D head itself introduces the possessor, though little would change if we were to employ a separate POSS head.

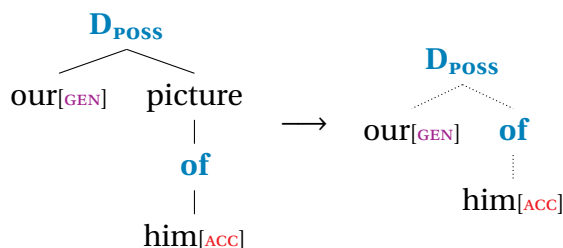
(21) a picture of him (internal argument)



(22) our picture (external argument)

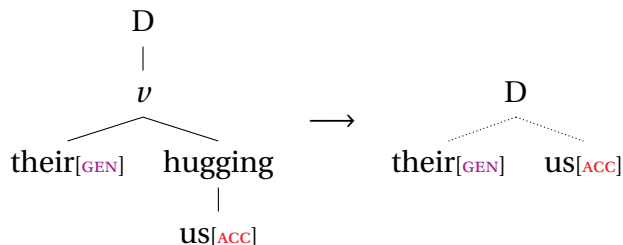


(23) our picture of him (two arguments)



Notice how in (21) the preposition shields the sole argument DP from the head of the containing DP, allowing it to remain accusative. (The fact the D head has *of* as a daughter technically violates its daughter string language, but this will be fixed momentarily.) The object is also shielded (23), though it is not crucial in this case. Recall that we can also have an accusative object together with a genitive subject in certain gerunds, labeled ‘hybrid’ earlier, as depicted in (24) below. Here, I assume that the subject originates as the specifier of νP and raises (movement features omitted for simplicity).

(24) their hugging us (hybrid gerund)



Unlike the preceding examples, the accusative DP is a daughter of D , not of P . This is why the daughter string language must be set up as it is. While we could in principle add V as a domain node, this would interfere with the analysis of unaccusative and raising verbs, since the subject would be incorrectly shielded from finite C . With either a more surface oriented representation or an analysis with multiple tiers, this option could become viable again.

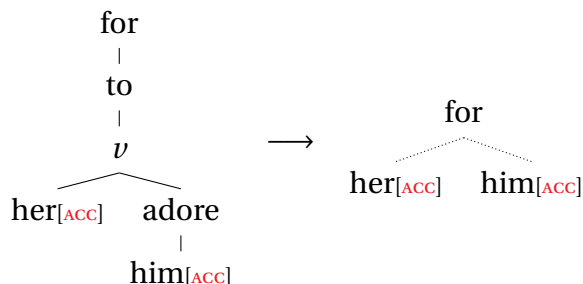
Now we must deal with PPs and embedded clauses (finite or otherwise). We determined that only accusative DPs are possible in these contexts, which means that their daughter string language is even simpler than the others.

(25) Daughter string grammar for P/C_{INF}

$$G^+ = \{\times \cdot D_{[ACC]}, D_{[ACC]} \cdot D_{[ACC]}, D_{[ACC]} \cdot \times, \times \cdot \times\}$$

This means that we can now handle *for*-to infinitives like the following:

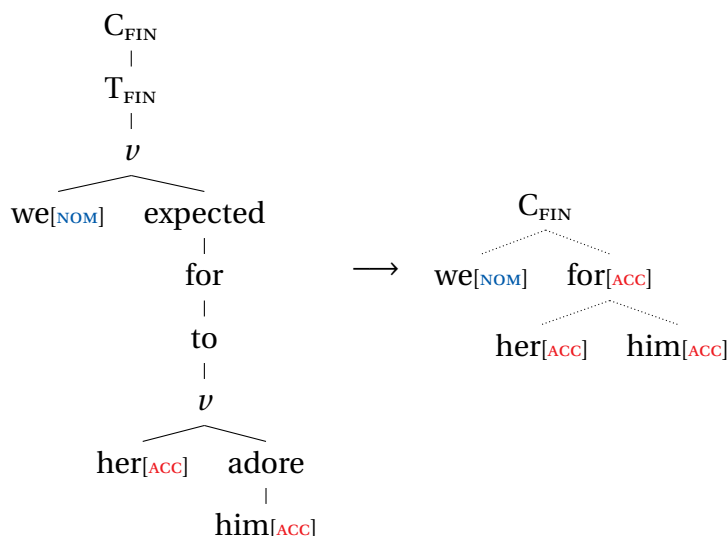
(26) *for her to adore him* (non-finite CP)



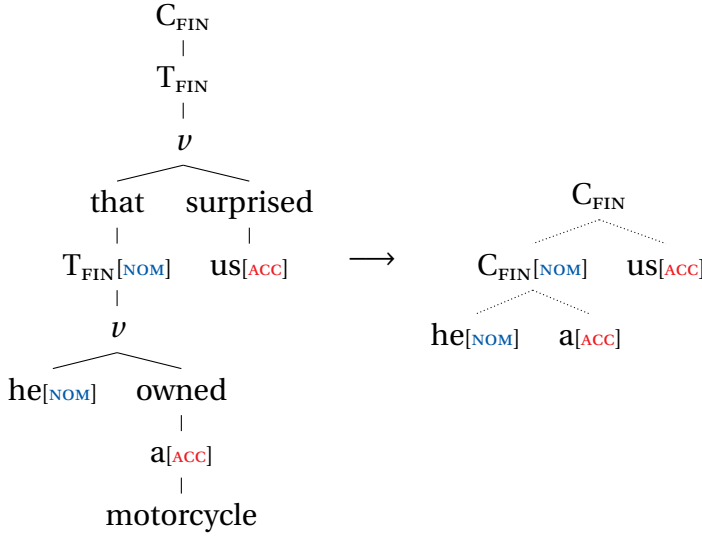
In the traditional generative analysis, we would say that *for* assigns accusative case to the subject, while the verb assigns it to the object, while in the present analysis, there is no such distinction. As before, accusative simply fills all the gaps where no other case is available. This is what it means to be a default case in the present system.

Now we must decide what to do about C/P daughters of other nodes on the tier. The most straightforward option is to say that they **ABSORB** the case that would otherwise be assigned in that position. That is, they are marked for case, but show no morphological realization, just as lexical DPs do not. (Later, we will consider circumstances where certain elements are invisible in the daughter string language rather than filling a case slot.) The following examples demonstrate absorption of accusative by a clausal complement, and of nominative by a clausal subject (movement features omitted for simplicity).

(27) *We expected [CP for her to adore him]* (non-finite CP)



(28) That he owned a motorcycle surprised us



In implementation, the main substantive change to be made is in the MG lexicon, which must allow C and P heads to bear case features just like D heads. In contrast, the changes to the TSL grammar are basically mechanical in nature. This is because our grammars are really meta-grammars, where $D[NOM]$ stands for any element of category D bearing the case feature **NOM**. In principle, we just need to duplicate each bigram to allow $C[NOM]$ and $P[NOM]$ in the same positions, and likewise for accusative (it is not clear that these categories can occur in a genitive position). However, the result is considerably harder to interpret than the daughter string grammars provided above. Therefore, I introduce a further notational shortcut: the label **NOM** stands in for any nominative XP, regardless of its other features, and likewise for **ACC** and **GEN**. Putting all of this together, we arrive at the following TSL grammar for case in English.

(29) Case tier for English:

- Project: {D, C, P}
- Daughter string grammars:
 - C_{FIN} :** $\{\times \cdot \text{NOM}, \text{NOM} \cdot \text{ACC}, \text{ACC} \cdot \text{ACC}, \text{NOM} \cdot \times, \text{ACC} \cdot \times, \times \cdot \times\}$
 - D:** $\{\times \cdot \text{GEN}, \text{GEN} \cdot \text{ACC}, \text{ACC} \cdot \text{ACC}, \text{GEN} \cdot \times, \text{ACC} \cdot \times, \times \cdot \times\}$
 - C_{INF}/P :** $\{\times \cdot \text{ACC}, \text{ACC} \cdot \text{ACC}, \text{ACC} \cdot \times, \times \cdot \times\}$

At this point, we have completed the formalization of our descriptive generalizations. To reiterate, this is not meant to be a complete analysis of English by any means, and there are several obvious deficiencies. Next, we consider the limitations of the analysis, and how the perspective it embodies naturally leads to certain formal questions which we will explore in subsequent chapters.

3.6 Some technical problems and their solutions

To my mind, there are three major problems with the analysis of English presented in the preceding section:

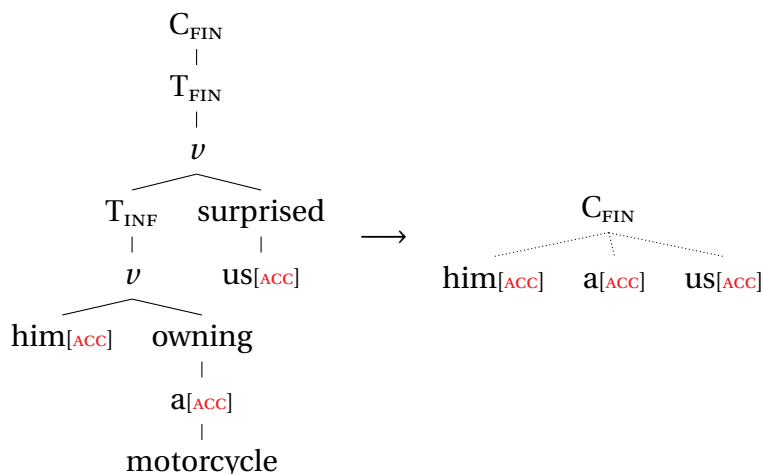
1. Non-finite TP subjects
2. Adjunct PPs
3. Pseudo-passives

Interestingly, each problem exposes a contradiction in visibility requirements, rendering the approach in the previous section inadequate. However, the solution to each is completely different: Problem 1 requires a version of the left branch condition, Problem 2 motivates tier projection in the daughter string language, and Problem 3 points towards a different formalization of the basic generalizations. Let us briefly treat each of these in turn.

3.6.1 Non-finite TP subjects, left branch visibility

When translating the English descriptive generalizations into semi-formal constraints, I noted that the translation from “highest DP in some domain” to “first daughter of a domain node on the tier” is not quite correct. This is because our common notion of “highest” is based on c-command, but as discussed in Chapter 2, tree tiers do not preserve c-command relations. In the above analysis, this shows up when a non-finite TP appears as a subject rather than a complement. An example which demonstrates the problem is given below.

(30) [TP him owning a motorcycle] surprised us



Here, the arguments of the embedded clause are combined with those of the surrounding clause, resulting in a daughter string with three accusative D heads. This contradicts the daughter string language of finite C, which says that the first should be nominative. In reality, none of the DPs in such a structure can be nominative—not the embedded subject, but also not the matrix object. If anything, the head of the infinitive TP should absorb nominative

case, just as a C or P would. The reason we did not do this in the first place is that it in complement position infinitive TPs (or at least those headed by *to*) should clearly be transparent; otherwise raised subjects would be accusative.

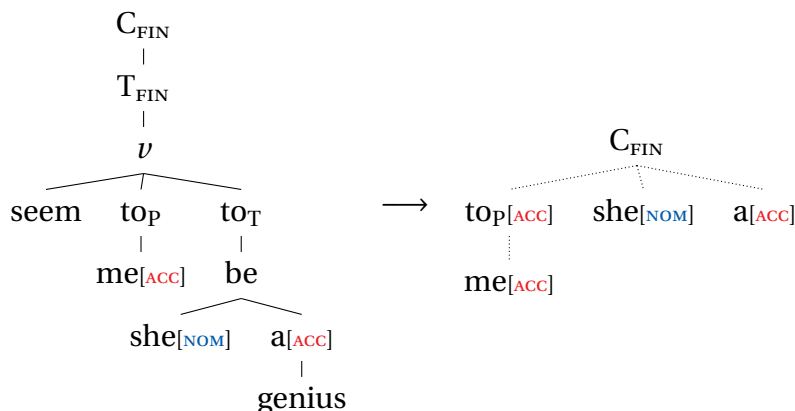
There are various solutions to this puzzle. For example, perhaps infinitive subjects really are DPs, or perhaps sentential subjects do not actually exist, i.e. they are always topics. Pursuing this issue would take us down well-trodden roads (see Davies and Dubinsky 2009; Lohndal 2014 and references therein). More generally, it exposes a fundamental aspect of syntax which the tree tier model does not handle: sometimes, specifiers (and adjuncts) are opaque even though complements are not. Because the tier projection function cares only about the label of a node, such distinctions can not be made.

Note that the problem is not at all specific to tier-based models, and seems to be related to the geometry of syntactic representations (c.f. Kayne 1984; Graf and De Santo 2019; Adger 2025). Accordingly, I adopt a slightly different model for agreement starting in Chapter 6, in which we project tiers from paths which are guaranteed not to enter a left branch. It seems plausible that the analysis of case could also be recast in that system, or the tree tier model adjusted to achieve the same effect, perhaps by marking specifiers and adjuncts so that they can be identified (cf. Graf and De Santo 2019). In the meantime, there are many aspects of case which *are* very naturally modeled with tree tiers, which we do not wish to lose. Also, we usually want to project the head of a left branch on the tier anyway as it tends to be of category D/C/P, so in practice this particular problem does not show up too often.

3.6.2 Adjunct PPs, tier projection in the daughter string

Depending on where adjuncts are thought to attach in the dependency tree, adjunct PPs could interfere with the case assignment rules as currently written. For example, if a PP adjunct appears as the first daughter of C_{FIN} , it will absorb nominative case, leaving the subject with accusative, which is clearly not right. Although I have not specified how adjunction is to be modeled in the MG dependency tree since we will not deal much with adjuncts in this work (see Hanson 2025a for how I envision this), the same problem can actually be demonstrated within the current system with raising across a *to*-dative experiencer, which is plausibly an optional object in a structurally superior position. The structure in question is shown below.

(31) She seems [_{PP} to me] [_{TP} to be a genius].



On the tier, we have an accusative D to the left of a nominative, which is clearly disallowed. At the same time, we need P heads to be present on the tier in order to shield their objects from being dispersed among the DPs in the surrounding domain (this includes the present structure). What we would really like to do is to project the P heads on the tier initially for their shielding effect, and only subsequently ignore them. Conveniently, tree tiers as defined by Graf and Kostyszyn (2021) already allow this: the daughter string grammar may itself be a TSL-2 grammar. By upgrading from SL-2 to TSL-2, we can ignore the P heads in the daughter string language, then use the exact same set of bigrams as before. The daughter string grammar of C_{FIN} in this case would be:

(32) **Daughter string grammar for C_{FIN} (revised)**

$$\Sigma = \{D, C, P\}$$

$$T = \{D, C\}$$

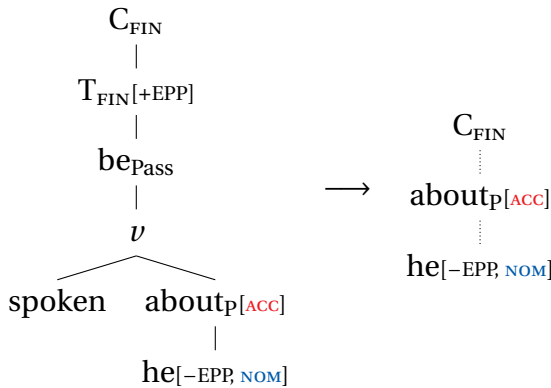
$$G^+ = \{\textcolor{blue}{\times} \cdot \textcolor{blue}{\text{NOM}}, \textcolor{blue}{\text{NOM}} \cdot \textcolor{red}{\text{ACC}}, \textcolor{red}{\text{ACC}} \cdot \textcolor{red}{\text{ACC}}, \textcolor{blue}{\text{NOM}} \cdot \textcolor{black}{\times}, \textcolor{red}{\text{ACC}} \cdot \textcolor{black}{\times}, \textcolor{black}{\times} \cdot \textcolor{black}{\times}\}$$

Admittedly, something more is needed if we want to maintain the possibility of PP subjects, but the basic idea is quite general: whenever we want to preserve a node in the tree tier, then ignore it in the daughter strings, we can do so. Again, we will not be dealing much with adjuncts (which are troublesome in many ways, regardless of syntactic framework), so I will continue to use SL-2 daughter string grammars in the following chapters, but it is good to keep in mind that there is always an opportunity for a second tier projection.

3.6.3 Pseudo-passives, an alternative analysis

Ordinary passives are not a problem for the current analysis following the conventional assumption that the agent is missing from the structure (or otherwise rendered invisible). As in other analyses, passives and unaccusatives are unified with raising: the highest DP is the target of both EPP-movement and nominative case assignment. Pseudo-passives, however, are a problem, as the subject remains shielded by the PP in the dependency tree. This is demonstrated below.

(33) He was spoken about (secretly)



Once again, there are a variety of possible solutions. To me, what this data suggests is that nominative case in English really does target the canonical subject position in Spec-TP. In this case, rather than targeting the highest DP in the dependency tree, it should target the highest EPP-mover, which is guaranteed to end up in this position. This would work for the present example as well as those in which some EPP-movers are accusative; the latter are either not the highest in the finite clause (19) or otherwise protected by another domain node (27). The downside, perhaps unavoidable, is that the rest of the analysis must be restructured: P heads no longer create a case domain, except perhaps when they themselves move to Spec-TP, genitive assignment should apply to movers to Spec-DP, and so on.

In contrast, in languages in which case is closely related to base argument configurations, the above analysis as stated is probably much closer to the truth. This can be seen in Icelandic, where assigning dative case to the subject causes nominative to be displaced to the object, and in Japanese, where scrambling has no effect on case. This is why I stated at the outset that I chose the present analysis not so much because I think it is the right analysis for English, but because it is the start of the right analysis for many languages, in which the relevant patterns are both richer and more clear.

3.7 Discussion

Now that we have seen the proposed system in action, there are a few big picture questions I would like to address. First, how does the approach to case proposed here compare to the constraints on movement studied in the previous chapter? Second, what does the model have to say about the typology of case assignment? Finally, do we need multiple case tiers, and if no, how many?

3.7.1 The nature of the constraints

While the TSL-2 model of case proposed in this chapter is in some ways very similar to the approach to movement described in the previous chapter, there is a crucial difference, which is we now make use of the order among tier sisters, whereas before we did not. In Chapter 4, I will make even heavier use of this information in order to derive variation in alignment as well as several kinds of differential argument marking. The idea, in a nutshell, is that natural language has access to this information, and so it is no surprise that it uses it at least some of the time.

In Section 3.5, I noted that tier sister order here serves as a proxy for c-command, and while this does by and large seem to work, it is theoretically possible that this could break down. Furthermore, past work on movement has shown no particular need for it; for example, the TSL-2 grammars for both single and multiple *wh*-movement work just fine if the tier sisters are randomly shuffled. This property, called permutation closure, would allow us to further restrict the range of possible TSL-2 grammars (Thomas Graf, p.c.). So, there might be a principled difference between case and movement, or these could be signs that the present approach is not quite right.

In the future, it would be useful to determine whether the case analyses presented here could be translated into a version of the command-string based system, such as that introduced in Chapter 6 for modeling agreement. This might resolve the conflicts just mentioned. Alternatively, we might find that we really do need a direct relation between case assigners and assignees as well as a relation between assignees, currently the tier sister relation. If so, some sort of hybrid system, combining tier-based dominance (from the tree tier system) and tier-based command (from the command string system), might do the trick. I leave these investigations for future work.

3.7.2 Consequences for typology

I wish to draw attention to several aspects of the TSL-2 model of case just sketched, which we will explore in detail in Chapter 4.

First, the model provides two basic parameters for manipulation: i) the tier alphabet, which determines which items are visible, and ii) the bigrams of the daughter string grammars. Each of these is expected to vary somewhat across languages, and between different tiers and daughter strings within the same language. Our goal is to show that the obvious ways that they may differ aligns closely with the variation that we see.

Second, we have considerable control over the relationship between a domain node and two adjacent DPs on the tier, but our ability to coordinate elements which do not stand in such a relationship is extremely limited. Accordingly, we will show how many types of conceptually simple yet linguistically unnatural patterns are ruled out on the grounds that they are not TSL-2. Of course, not every unattested pattern can be ruled out in this manner. Even so, we have a strong hypothesis for what constitutes a possible case pattern, which plausibly derives from some limitation of the underlying cognitive system (Graf 2022b). This in turn clarifies which gaps *must* be accounted for by other factors, computational or otherwise.

Finally, the above points apply equally to agreement, even though the details of the model are somewhat different. The TSL model for agreement, based on command strings, is laid out in Chapter 6, and the typological predictions are explored in Chapter 7.

3.7.3 How many case tiers?

Recall from Chapter 2 that we generally assume each logical dependency to have a distinct tier and constraints. That is, while individual dependencies are TSL, the combined grammar is (at least) MTSL. The primary exception has to do with local dependencies, including selection but also presumably lexical case. Since all elements are visible in a local dependency, the tier is always the same, and the constraints can be combined into a single (SL) grammar.

This distinction is relevant because it is not always obvious what counts as a separate dependency. In most of the examples in Chapter 2—*wh*-movement, EPP-movement, NPI-licensing, etc.—it was fairly clear that we were dealing with separate dependencies, and that one tier for each is sufficient. This is considerably less clear for case. In this chapter, I presented an analysis of case in English which handled three cases with a single tier, though I also mentioned that this is probably not sufficient for other languages. This does not invali-

date the typological claims made in Chapter 4, but it suggests some urgency in determining to what degree case involves multiple tiers, more so than for other syntactic phenomena.

To get a feel for the problem, consider the interaction between structural and lexical case, which necessarily occupy distinct tiers, as the former is non-local while the latter is local. In many languages a DP receives the structural case appropriate for its position iff it has not been assigned lexical case by its selecting predicate. The structural and lexical case tiers must coordinate to ensure that this happens, but without knowing exactly what the other looks like—all each can do is rule out certain cases in certain positions. As the number of tiers increases, the need for such coordination will tend to increase.

How many tiers, then, do we need for case? Speaking strictly in terms of what is needed to capture the relevant generalizations, the minimum value of a single tier seems untenable, even though this is what one might predict based on the naïve assumption that case is a single morphological feature with multiple values. This can be seen as further justification for maintaining separate case features, as I have done here. At the opposite extreme, we might posit that every distinct case gets a unique tier, with the usual proviso that lexical cases may be combined on a single tier. It seems to me that the actual answer will be between these two extremes, but will depend somewhat on other criteria such as At present, my best guess is that that number of case tiers we need is somewhere between these two extremes. Plausibly, we need one tier per active case domain in a given language. This seems to be true of Japanese based on the analysis presented in Chapter 5, where certain facts about cross-clausal case assignment indicate the need for two non-local case tiers, corresponding to the verbal and nominal domains. In a nutshell, we need multiple tiers whenever certain domain nodes seem to be visible for some cases but not others.

It is not possible here to fully address the typological predictions of a multi-tier system for case, but Chapter 5 does lay out how I envision such a system working, inclusive of the coordination problem just mentioned. Similarly, the analysis of Hindi long-distance agreement in Chapter 8 incorporates multiple tiers, and suggests that the empirical phenomenon of parasitic agreement boils down to the same type of blind coordination among tiers.

3.8 Summary

In this chapter, I showed how case can be assigned to DPs using a system of tree tiers, updating the model of Vu et al. (2019) to work with MG dependency trees in a way which is extremely similar to the model of movement in Graf (2022c). The major innovation is that we make heavier use of relative positions of tier daughters, as is made possible by the TSL-2 daughter string languages. This model provides a concise and straightforward implementation of a configurational approach to case, which does not require distinct cases to have separate assigners, and also allows us to safely abstract away from many details of case theory to focus on one major question: how can the case of DPs be determined from their structural context? In the following chapters, I demonstrate the empirical coverage of the model, both in accounting for variation in case marking patterns across languages, and for dealing with the detailed facts of individual languages.

Chapter 4

The Typology of Case

4.1 Overview

In the previous chapter, I developed a model of case assignment using a system of tree tiers. In this chapter, I demonstrate how the proposed model accounts for many well-known and theoretically interesting patterns in the syntactic distribution of case marking. Following Graf (2022c), I emphasize how the natural parameters of the model—the tier projection and the constraints on the daughter strings—correspond closely to attested cross-linguistic variation. Specific phenomena of interest include case spreading, variation in alignment (accusative, ergative, etc.), interactions between lexical and structural case, domain effects caused by invisibility and blocking, and four types of differential argument marking. We also briefly discuss what we do *not* see, and why the model rules out such patterns.

The specific patterns covered here were chosen for several reasons. First, they are very common, and should therefore have a straightforward treatment. Second, they are theoretically interesting, and depending on one’s perspective, possibly surprising. Finally, many of them admit a range of subvariation which itself deserves an explanation. While far from exhaustive, the present study demonstrates in broad strokes that the TSL-2 model is not just up to the task, but brings insight into these case phenomena.

Before we begin, I should re-emphasize that the present study takes a configurational approach to case which is highly similar to DEPENDENT CASE THEORY (Marantz 2000; Baker 2015) as well as the earlier CASE IN TIERS framework of Yip et al. (1987). While I do not provide a direct comparison, I borrow several examples and their basic analysis from works in these tradition. It may also be possible to construct a more conventional analysis based on pairwise relations between case assigners and assignees, eschewing any relation between assignees. The present study can be interpreted as an argument against doing this, for the computational system has direct access to such information, and the patterns that we find imply that the system does use it.

The remainder of this chapter is organized as follows. The first five sections cover important case patterns described in the literature: spreading (Section 4.2), variation in alignment (Section 4.3), lexical case and its interactions with structural case (Section 4.4), variation in

This chapter is based on material from an early version of Hanson (2025b) “The Typology of TSL Case Assignment”. The final published version is not expected to see major changes in the analysis.

case domains (Section 4.5), and differential argument marking (Section 4.6). The remaining sections discuss non-existent case patterns and how they may be ruled out, whether by the TSL-2 hypothesis (Section 4.7) or otherwise (Section 4.8).

4.2 Spreading

One of the major motivations of Yip et al. (1987) was to account for case spreading, that is, situations where a single case is applied to multiple DPs in some domain. It is also a natural prediction of the TSL-2 model: all we need in the daughter string grammar is a bigram which duplicates the same case twice.

We already demonstrated spreading of accusative in the preceding section, though as Yip et al. (1987) note, Swedish would be a better example, since accusative is plausibly the default case in English.

- (1) Multiple accusative in Swedish (Yip et al. 1987)

Kungen gav honom henne (till maka).
the.king gave him.ACC her.ACC (for wife)

Similarly, Japanese allows multiple nominatives and genitives in several contexts.

- (2) Multiple nominatives in Japanese

- a. Stative object (Hiraiwa 2001a)

Mary ga eigo ga yoku dekiru.
Mary NOM English NOM well can.do
‘Mary can speak English well.’

- b. Possessor raising (Kuno 1973)

Yama ga ki ga kirei desu.
mountain NOM trees NOM pretty are
‘The mountains—their trees are pretty.’

- (3) Multiple genitives in Japanese

- a. Multiple possessors (Saito et al. 2008)

Taroo no Chomsky no hon
Taroo GEN Chomsky GEN book
‘Taroo’s book by Chomsky’

- b. Nominal subject/object (Saito et al. 2008)

yabanzin no Rooma no hakai
barbarian GEN Rome GEN destruction
‘the barbarians’ destruction of Rome’

By including **ACC·ACC**, **NOM·NOM**, or **GEN·GEN** in the relevant daughter string language (clause, VP, DP, etc.), we enable spreading without the need to posit additional (covert) case assigners. We also predict that spreading will continue if additional DPs are present, since a daughter string with three or more repetitions of, say, accusative, has exactly the same bigrams: $\{\times \cdot \text{ACC}, \text{ACC} \cdot \text{ACC}, \text{ACC} \cdot \times\}$. Put another way, an SL-2 grammar cannot distinguish “exactly two” from “two or more”.

While the literature rarely discusses such constructions, this appears to be true at least of possessor raising in Japanese (Kuno 1973).¹ Furthermore, restricting spreading to a *maximum* of two DPs requires at least TSL-3. This kind of counting is highly unusual in natural language, and TSL-3 can generate many other strange patterns that TSL-2 cannot (see Section 4.7). In absence of contradictory evidence, a TSL-2 analysis of spreading is preferred.

Another superficially similar phenomenon which may be amenable to the TSL-2 model is CASE CONCORD, in which the case of the DP is realized in multiple places, including the head noun, the determiner, adjectival modifiers, and perhaps other elements. An extreme example from Panjima (Pama-Nyungan), with five concordial elements, is shown below.

- (4) Ngatha wiya-rna ngunha-yu maripa-yu paka-lalha-ku nharniwalk-ku
 I.**NOM** see-PST that-**ACC** man-**ACC** come-PRF-**ACC** hither-**ACC**
 warrungkamu-la-ku
 morning-LOC-**ACC**

‘I saw that man who came this way this morning.’ (Blake 2001, ch. 4)

There are several ways this might be implemented. The direct approach involves duplicating the bigrams of the spreading grammar to allow iteration of each case. The trouble with strategy is that the case shared among the concordial elements not arbitrary, but rather determined by the case of entire DP. One possible solution is to *split* the daughter string language for D on the case tier into several, one for each case.

- (5) Schematic grammar for case concord

D[**NOM**]: $\{\times \cdot \text{NOM}, \text{NOM} \cdot \text{NOM}, \text{NOM} \cdot \times, \times \cdot \times\}$

D[**ACC**]: $\{\times \cdot \text{ACC}, \text{ACC} \cdot \text{ACC}, \text{ACC} \cdot \times, \times \cdot \times\}$

D[**DAT**]: $\{\times \cdot \text{DAT}, \text{DAT} \cdot \text{DAT}, \text{DAT} \cdot \times, \times \cdot \times\}$

etc.

A better approach might be to treat case concord as a kind of agreement, since person/number/gender features are often shared as well. Furthermore, the class of items involved is somewhat different, including nouns and adjectives but excluding possessors and arguments (e.g. the possessor of an accusative DP does not become accusative). While the basic idea remains the same, the tier for case ASSIGNMENT would be completely separate from those for case CONCORD; see Hanson (2024b) for a possible approach to the latter. This

¹Numeral quantifiers and PP modifiers of nouns also take the *no* clitic, which would seem to imply an unlimited number of genitives. But *no* can also be an adnominal copula, and it is difficult to tell apart these two functions.

can be seen as an example of how tiers are best thought of as corresponding to syntactic dependencies rather than classes of morphological features.

To briefly summarize, the TSL-2 model predicts the existence of iteration. The appears to be borne out in several ways in the realm of case, mirroring its appearance in long-distance harmony, multiple *wh*-movement, and person/number/gender concord in the DP. However, we cannot reasonably expect every such pattern to occur in each of these domains. In particular, there are TSL-2 patterns which are highly characteristic of case but which are not nearly as robustly attested elsewhere. We consider some of these in the following sections.

4.3 Alignment

Case marking correlates imperfectly with both semantic role and grammatical function. In line with the configurational view adopted above, I show that many of the most common patterns correspond exactly to minor differences in the structure of the SL-2 daughter string language on a case tier. Note that many languages employ multiple alignments, controlled by various structural and lexical factors, which we abstract away from here.

Following common convention, I refer to the sole argument of an intransitive verb as S, the “more agent-like” argument of a transitive verb as A, and the “less agent-like” argument as O (cf. Bickel and Nichols 2008). The case of S is usually the (literally) unmarked case, which we call NOMINATIVE. When A and S share a case and O receives a different case, we call the case of O ACCUSATIVE. This is known as accusative alignment, and is the dominant pattern in most Indo-European languages. It is also common for S and O share a case to the exclusion of A. We call the case of A ERGATIVE, and the overall pattern is known as ergative alignment. This is the dominant pattern in Shipibo (Panoan family).

(6) Shipibo (Baker 2015)

- a. Maria-nin-ra ochiti noko-ke.
Maria-ERG-PRT dog find-PRF
‘Maria found the dog.’
- b. Maria-ra ka-ke.
Maria-PRT go-PRF
‘Maria went.’

For the languages under discussion, I assume a consistent structural relationship such that A is higher than O. I further assume following Baker (2015) that ergative case in Shipibo is assigned to the higher of two DPs (the A argument) in the same clause.² The basic grammar for ergative alignment is therefore nearly identical to that for accusative. The key difference is that the ‘new’ case of the transitive structure is introduced to the left of nominative case.

(7) Grammar for ergative alignment

$$G^+ = \{ \times \cdot \text{NOM}, \times \cdot \text{ERG}, \text{ERG} \cdot \text{NOM}, \text{NOM} \cdot \times \}$$

²It is likely that not every language with a case called ‘ergative’ works like this, which Baker acknowledges. See Butt (2006, ch. 6) for additional discussion. In Section 4.5, we examine another type of ‘ergative’ system.

It is also possible for S, A, and O to each have a distinct case (‘tripartite’ alignment), or for them to all share the same case (‘neutral’ alignment). This is summarized in the table below.

(8) Common case alignments

Alignment	Case of		
	S	A	O
Accusative	NOM	NOM	ACC
Ergative	NOM	ERG	NOM
Tripartite	NOM	ERG	ACC
Neutral	NOM	NOM	NOM

We see that every logical possibility for the cases of the canonical transitive verb is realized: either A or O may have a special case, or both, or neither. Furthermore, each of these requires only a few bigrams to be changed compared to our accusative and ergative grammars. The neutral grammar is the same as what we called ‘spreading’ in §4.2, and the tripartite grammar is given below.

(9) Grammar for tripartite alignment

$$G^+ = \{\times \cdot \text{NOM}, \times \cdot \text{ERG}, \text{ERG} \cdot \text{ACC}, \text{NOM} \cdot \times, \text{ACC} \cdot \times\}$$

Note that some authors, such as Bittner and Hale (1996), distinguish even more systems. For example, internal and external arguments of intransitive verbs may behave differently, which is a major factor in certain dialects of Basque. Additionally, alignment may change according to tense or aspect, as in Hindi. From the present perspective, such “alignments” usually involve multiple domain nodes, and will be addressed in Section 4.5.

Now, we ask what happens when there are more than two arguments. Following Bickel and Nichols (2008), I refer to the two objects of a ditransitive verb as G (for the more goal-like argument) and T (for the more theme-like argument); ditransitive subjects are rarely distinguished from transitive subjects, so they are still A. It is common for either G or T to receive the same case as O, while the other receives a new case. For example, G can be DATIVE, and T aligned with O. This occurs both in languages with accusative alignment, such as German and Japanese, as well as languages with ergative alignment, such as Ingush (Nakh-Daghestanian).

(10) Japanese ditransitive verb

Jin ga Yumi ni hon o ageta.
 Jin NOM Yumi DAT book ACC gave
 ‘Jin give Yumi a book.’

(11) Ingush ditransitive verb

Aaz cynna mashen jelar
 1SG.ERG 3SG.DAT car.NOM gave

‘I gave him/her a car’ (Bickel and Nichols 2008)

Taking the default linear order to indicate c-command among base argument positions, the grammar for such systems are given below.

(12) Accusative alignment with dative G

$$G^+ = \{\times \cdot \text{NOM}, \text{NOM} \cdot \text{ACC}, \text{NOM} \cdot \text{DAT}, \text{DAT} \cdot \text{ACC}, \text{NOM} \cdot \times, \text{ACC} \cdot \times\}$$

(13) Ergative alignment with dative G

$$G^+ = \{\times \cdot \text{NOM}, \times \cdot \text{ERG}, \text{ERG} \cdot \text{NOM}, \text{ERG} \cdot \text{DAT}, \text{DAT} \cdot \text{NOM}, \text{NOM} \cdot \times\}$$

It is also possible for T to have a special case and G aligned with O, or for all three to be aligned, with various relationships to S/A. On top of this, some languages may employ low goals rather than high goal, or perhaps include both structures. While we cannot do justice to the full typology here, what is clear is that as with transitives, several well-attested systems correspond to simple manipulations of the TSL-2 grammar: is there a third basic case and does it occur before or after the case of O?

We could continue in this fashion by asking what happens when there are four DP arguments. Although simplex four-place predicates with DP arguments are rare (perhaps nonexistent), they can be constructed by valency increasing operations such as morphological causatives. Unfortunately, as with spreading, the literature on case rarely discusses such data. For instance, not a single example of a causative of a ditransitive could be found by searching all index entries for both “causative” and “ditransitive” in the 900-page Oxford Handbook of Case (Malchukov and Spencer 2008). However, as we will see in Chapter 5, such constructions are possible in Japanese, and that both datives are plausibly structural, or at least ambiguous between structural and lexical dative.

(14) Causative of ditransitive

Ken ga Jin ni Yumi ni hon o agesaseta.

Ken **NOM** Jin **DAT** Yumi **DAT** book **ACC** gave.CAUS

‘Ken made/let Jin give Yumi a book.’

The grammar proposed there, shown below with extraneous details removed, involves spreading of dative.³

(15) Verbal cases in Japanese

$$G^+ = \{\times \cdot \text{NOM}, \text{NOM} \cdot \text{ACC}, \text{NOM} \cdot \text{DAT}, \text{DAT} \cdot \text{DAT}, \text{DAT} \cdot \text{ACC}, \text{NOM} \cdot \times, \text{ACC} \cdot \times\}$$

While further investigation is clearly needed, our survey of alignment clearly supports the idea that the various structural cases available in a given domain can be assigned according to a TSL-2 grammar. There are, of course, alternatives that could fit the readily available data. In particular, we could employ a more traditional generative analysis, in which each case is associated with a unique functional head: T, Aspect, Voice, Appl, etc. But then it would be a coincidence that so many case patterns can be straightforwardly derived with TSL-2 and very coarse case domains.

³The grammar is meant to apply to arguments of canonical VPs. Stative VPs are assumed to be defective, and invisible on the relevant tier.

4.4 Lexical case

In the preceding sections, we focused mostly on the structure of the constraints. Now, we begin to consider the contents of the tier alphabet as well.

In Section 3.5, we showed how a clausal subject fills the nominative slot in English, which we called case absorption. When a lexically case-marked DP occurs in a particular case position, it is possible for the corresponding structural case to instead be **DISPLACED** to the next available DP. This occurs with lexical datives in Icelandic.

(16) Case displacement in Icelandic (Yip et al. 1987)

- a. Siggi leyndi konuna sannleikanum.
S.**NOM** concealed the.woman.**ACC** the.truth.**LDAT**
'Siggi concealed the truth from the woman.'
- b. Siggi sagði barninu söguna.
S.**NOM** told the.child.**LDAT** the.story.**ACC**
'Siggi told the child the story.'
- c. Barninu batnaði veikin.
the.child.**LDAT** recovered.from the.disease.**NOM**
'The child recovered from the disease.'

When **LDAT** occurs in the lowest (oblique) position, the subject and first object are unaffected. But when it marks in the first object, accusative is displaced to the second object, and when it marks the subject, nominative is displaced to the object. Icelandic is a particularly good example of the latter phenomenon, as it is clear that the lexically-case marked subject is in fact in the usual subject position (Zaenen et al. 1985).

In the present analysis, this can be achieved if we make lexical datives invisible on the tier controlling structural case in the clause. That is, the tier alphabet contains T and any D marked for structural case (nominative, accusative, and perhaps structural dative) but not lexical dative. As a result, the tiers for the above examples look the same as they would for an ordinary transitive/intransitive clause.

There is a complication, which is that any DPs inside the invisible lexical dative will be strewn among the arguments of the containing clause. The solution I adopt capitalizes on the fact that a daughter string language in the tree tier system may be TSL-2, rather than SL-2 as we have assumed so far. In this case, we include lexical datives in the tree tier, but ignore them in the daughter string language. This is illustrated for (16b) in Figure 4.1. In Chapter 5, I use the same technique to factor out PP adjuncts (lexical datives do not require this treatment due to several quirks of the grammar of Japanese). We might also use it for PP arguments that induce case displacement, to the extent that such PPs can be distinguished from DPs with lexical case.

As a final remark, though we have not specified how lexical dative itself is assigned in Icelandic, lexical case is generally held to be assigned locally to arguments of specific predicates. This means that a simple SL grammar will do, as with other local dependencies (cf.

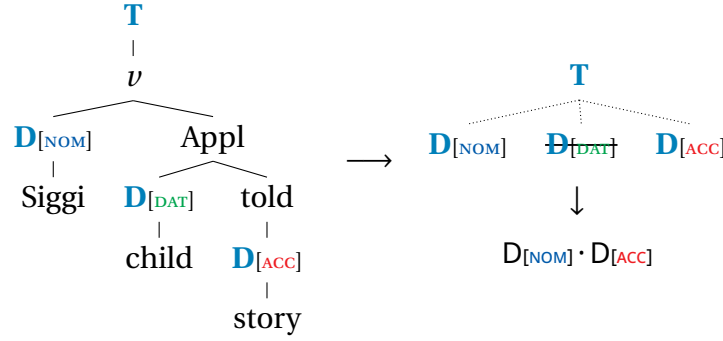


Figure 4.1: Derivation of case displacement. Lexical datives appear on the tree tier, but are ignored in the daughter string language. Nominative and accusative are adjacent as usual.

Hanson 2024c). More properly, we should say lexical case occupies the tier which contains everything. But as we have seen, it is still formally interesting due to its interaction with cases assigned on non-trivial tiers. The following section discusses some other examples of the effect of differences in visibility on a case tier.

4.5 Domains and their interactions

As outlined in Chapter 3, it is normal to have several case domains, which might include TP, VP, DP, and perhaps others depending on the language. In particular, it is common for case marking to be controlled partly by aspect, as in Hindi.

(17) Alignment split by aspect in Hindi (Mahajan 1990)

- a. Raam roṭii khaataa thaa.
Raam.**NOM** bread.**NOM** eat.IPFV be.PST
'Raam ate bread (habitually).'
- b. Raam-ne roṭii khaayii.
Raam-**ERG** bread.**NOM** eat.PFV
'Raam ate bread.'

Based on this data, we would say that imperfectives have neutral alignment while perfectives are ergative. (Hindi also has differential object marking, which is covered in §4.6.) Leaving aside the details, let us assume that the Asp(ect) head, realized by an auxiliary only in the imperfective, controls alignment according to the following grammar:

(18) Split alignment in Hindi

- Project: {Asp, D}
- Daughter string grammars:
Ipfv: { $\times \cdot \text{NOM}, \text{NOM} \cdot \text{NOM}, \text{NOM} \cdot \times$ }
Pfv: { $\times \cdot \text{NOM}, \times \cdot \text{ERG}, \text{ERG} \cdot \text{NOM}, \text{NOM} \cdot \text{NOM}, \text{NOM} \cdot \times$ }

In reality, it is more likely that neutral and ergative alignment are associated with Asp and T or vice versa (Thomas McFadden, p.c.), but without delving into the details, it is difficult to distinguish between these alternatives. To get a sense of how this works, let us consider another language with ergative case: Basque.

In Basque, transitive clauses show ergative alignment, but the case of an intransitive subject varies. In western/central dialects, the basic generalization is that a lone external argument receives ergative case, while a lone internal argument receives absolutive (=nominative) case, a pattern which is sometimes called ‘Split S’.

(19) Split S in Basque (Berro and Etxepare 2017)

- a. Jon-ek dantza-tu du.
Jon-**ERG** dance-PRF AUX
‘Jon has danced.’ (unergative)
- b. Jon eror-i da.
Jon.**NOM** fall-PRF AUX
‘Jon has fallen.’ (unaccusative)

By testing a variety of constructions, it is possible to show that the availability of ergative case is closely tied to a DP being in the domain of finite T (Rezac et al. 2014). For example, embedded finite clauses can have ergative subjects, whereas in gerunds, which are non-finite, ergative marking is lost.

(20) Ergative is lost in gerunds (Rezac et al. 2014)

- a. [Katu-ek sagu-ak harrapa-tu dituzte-la] ikusi dut.
cat-PL.**ERG** mouse-PL.**NOM** caught AUX-that seen AUX
‘I saw that the cats caught the mice.’
- b. [Katu-ak sagu-ak harrapa-tzen] ikusi ditut.
cat-PL.**NOM** mouse-PL.**NOM** catch-ing seen AUX
‘I saw the cats catch the mice.’

We can derive this pattern if we assume that finite TP and VP are case domains and that finite T assigns ergative while V assigns nominative. Internal arguments will always be shielded from finite T since they originate inside VP. For present purposes, I assume that non-finite T also assigns nominative just like V. This is illustrated in Figure 4.2. I leave open whether the domain nodes should absorb case, or whether they should be ignored using the technique from Section 4.4.

The above example involves intervention by a domain node on the tier, but it is also possible for a domain node to be invisible in certain contexts. This is the basis of the analysis of long-distance case assignment in Chapter 5. While I cannot go into detail here, the basic idea is that certain lexical predicates optionally select an embedded complementizer which is invisible on the relevant tier. Though it is perhaps more common to analyze such phenomena as involving a truncated clause structure, the TSL approach allows us abstract away from the

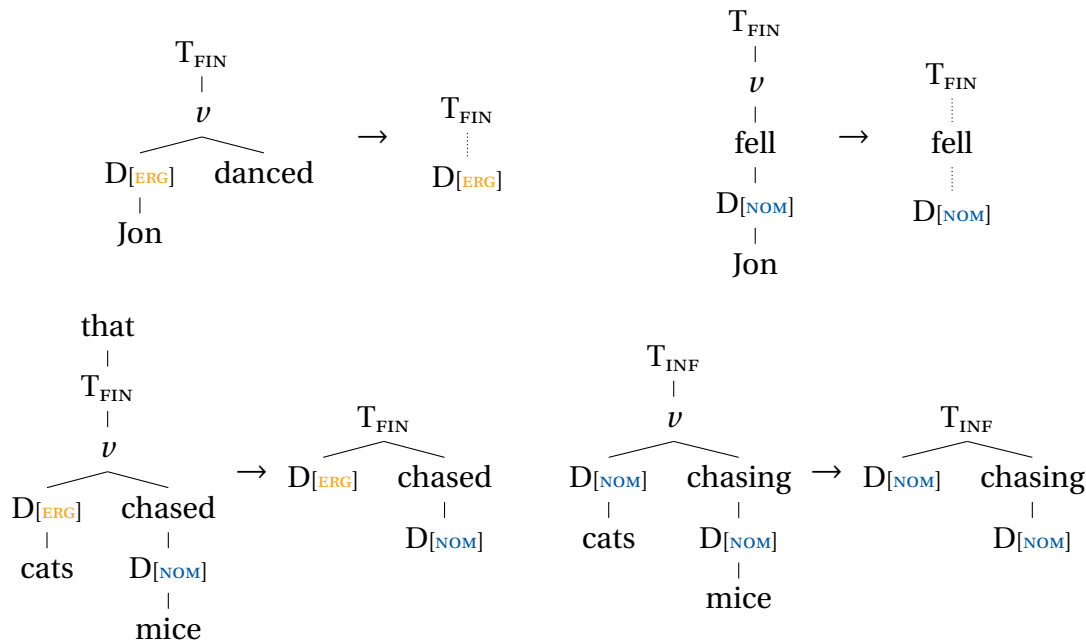


Figure 4.2: Ergative in Basque is available only in the domain of finite T. From top left to bottom right: unergative, unaccusative, transitive finite, transitive gerund.

precise nature of the structural difference. This is helpful for Japanese since the particle *to* appears extremely high in the extended CP (Saito 2015), yet is still present under long-distance assignment of accusative.

To summarize what we have seen so far, we can derive many of the most common variations in case marking simply by manipulating the contents of the tier alphabet, the number of cases, and their positions in the bigrams of the daughter string grammars. In the next section, we consider our final variable, which is to utilize features of DPs other than case in our grammars.

4.6 Differential argument marking

Differential argument marking (DAM) refers to phenomena in which some morphological marking of a DP, including but not limited to case, is only sometimes present; see Witzlack-Makarevich and Seržant (2018) for an overview. DAM subsumes both differential object marking (DOM) as well as differential subject marking (DSM), and tends to be linked to discourse and semantic factors, which themselves could involve the argument itself or a co-argument. Here, we show that the TSL-2 model derives this four-way typology.

We begin with a classic example of DOM, as illustrated with accusative case in Sakha (Turkic). In Sakha, accusative case marking is associated with a definitive/specific interpretation as well as movement out of the VP, as diagnosed by the position of manner adverbs.

- (21) Sakha DOM (Baker and Vinokurova 2010)

- a. Masha salamaat-*(y) tūrgennik sie-te.
Masha porridge-**ACC** quickly eat-PST.3SG
'Masha ate the porridge quickly.'
- b. Masha tūrgennik salamaat-(#y) sie-te.
Masha quickly porridge-**ACC** eat-PST.3SG
'Masha ate porridge quickly.' (**ACC** on 'porridge' only with contrastive focus)

Baker and Vinokurova (2010) posit that accusative is assigned to the lower of two DPs in the domain of T (as we have been assuming), and that movement out of the VP is crucial in making the object visible for case marking. That movement is the trigger for DOM is not a foregone conclusion, as we discuss below, but let us adopt this view for now. As discussed by Graf (2022c), we can handle certain interactions between syntactic operations—in this case, movement—by referencing the MG features for one dependency in the TSL grammar for another; I explore this idea in more detail in Chapter 7. In this case, we assign accusative only to DPs which bear the feature for object shift in addition to appearing in the correct structural context. This is illustrated in the grammar below.

(22) Accusative DOM triggered by object movement

- Project: T_{FIN} , all D
- Grammar for T_{FIN} :
 $\{\times \cdot D_{[NOM]}, D_{[NOM]} \cdot D_{[oshift^-, ACC]}, D_{[NOM]} \cdot D_{[NOM]}, D_{[NOM]} \cdot \times, D_{[oshift^-, ACC]} \cdot \times\}$

Here, we make use of the fact that the subject and object are adjacent in the daughter string language in the construction of the constraints. If this type of constraint is generally possible, then we predict that movement of the object might also influence ergative case marking of the subject, a kind of DSM. Baker (2015, pp. 127-129) also gives several examples of this type. In addition, we predict that DSM could be triggered by an internal property of the subject, or conversely, DOM triggered externally by a property of the subject. As discussed by Daniel (2024), all four possibilities are realized. The set of examples she cites are given below.

(23) Typology of self/externally-driven DAM

Subtype	Locus of Controlling Property	
	Self	External
DSM	Kham, Folopa, Umpithamu	Eastern Ostyak, Ika, Kanuri
DOM	Hindi, Spanish, Turkish, Sakha	Ik

At least one cell in the typology is problematic for many approaches to case. For example, self-driven DSM is unexpected under Baker's assumptions, since subjects do not normally move for semantic/discourse reasons, while externally-driven DAM is generally unexpected

under functionalist theories (Baker 2015, pp. 129-130). From the present perspective, all four cells are equally easy to derive. Furthermore, movement need not be the trigger; it could also be agreement, or the underlying semantic/discourse feature, if we make it available in the syntax. Indeed, Hindi also has differential object marking (not shown previously), but unlike in Sakha it is not claimed to involve movement.

We can therefore provide an even simpler account than Daniel (2024) does: given that the subject and object appear in the same tier bigram, we can reference the properties of either in determining their cases. In future work it would be important to confirm whether global case splits (cf. Bárány 2017; Clem and Deal 2024), in which the properties of *both* the subject and object influence case or agreement, can be handled in the same manner. In principle this should be possible, but the data for such languages is complex enough that a closer look is warranted.

4.7 Non-existent case patterns

Having shown how many of the positive predictions of the TSL-2 model for case are borne out, we close this section by examining some non-existent patterns which are ruled out purely on formal grounds. Recall from Chapter 2 how many types of non-existent island constraints cannot be expressed using TSL-2. Now consider the following hypothetical case systems:

(24) Examples of non-TSL and non-existent case patterns

- a. **2x2 alignment:** the highest two DPs are nominative; the next two DPs are accusative; repeat
- b. **Anti-local dependent case:** the subject is nominative if there are one or two arguments and ergative if there are three or more; objects are always accusative
- c. **Balanced coordinate alignment:** when two clauses are coordinated, one must have accusative alignment and the other ergative, in either order
- d. **Rationed dative case:** up to two clauses may contain dative DPs; the rest use genitive instead
- e. **Rotating alignment:** the main clause is accusative; a singly embedded clause is ergative; a doubly embedded clause is accusative; etc.

All of these are straightforwardly ruled out due to requiring some non-TSL-2 mechanism, either in domain node identification or in the daughter string language:

- A window larger than two (24a, 24b)
- Global boolean logic (24c)
- Threshold counting (24d)
- Modulo counting (24e)

These examples were selected in part because they are conceptually straightforward yet extremely unnatural from a linguistic perspective. They are also simple in computational terms, all being regular over derivation trees, and with the exception of (24e), subregular. As

such, we have good reason to believe it is TSL-2 specifically that characterizes the class of possible case patterns; it is powerful enough to do the job, but also not overpowered.

One possible objection to this line of reasoning is that all of the above systems are clearly dysfunctional as a means of mapping DPs to grammatical positions, and are therefore already ruled out for this reason, irrespective of computational complexity. But this mapping is tenuous at best even in real languages. Already, we have seen many examples of grammatical roles being conflated under the same case, and of the same role being split. The mapping is also subject to extensive but not unlimited variation. There remains ample room for formal factors to play a role in explaining the typology, and the fact that TSL-2 makes approximately the right cut is unlikely to be a coincidence. This is especially true when we consider the broader picture, since seemingly unrelated phenomena such as movement and long-distance phonotactics also seem to obey the same restriction.

4.8 The overgeneration question

In the preceding section, we saw how many robustly attested case patterns are TSL-2, and in contrast, how many simple yet unattested patterns are not. We should also ask whether there are any instances of undergeneration and overgeneration. I am not aware of any attested case patterns which are clearly not TSL-2 (undergeneration), but it is easy to construct certain unattested patterns which are TSL-2 (overgeneration). For instance, nominative and accusative could alternate according to the following grammar:

$$(25) \quad G^+ = \{\times \cdot \text{NOM}, \text{NOM} \cdot \text{ACC}, \text{ACC} \cdot \text{NOM}, \text{ACC} \cdot \times, \text{NOM} \cdot \times\}$$

This pattern is only slightly different from (24a) in that it utilizes a window of size 2 rather than 3, but it is just as unnatural.

However, as was mentioned in Section 3.7.2, we cannot realistically expect every (T)SL-2 pattern to be realized in every type of linguistic dependency. Some, like iteration, are shared across many dependency types (see Section 4.2), while others show up only here and there. Indeed, although alternating patterns like the above are unnatural for case, they *can* be found in category selection along the complement spine—D selects N, which selects P, which selects D, and so on—as well as in syllable structure (Hanson 2024c). Another set of patterns seems to be characteristic of in feature-matching operations such as long-distance harmony and syntactic agreement, and to a lesser extent, movement. Case exhibits yet other another slice of the overall space of TSL-2 patterns.

Can we say anything more specific about the subclass of TSL-2 which is representative of case? I believe that we can. As mentioned at the outset, I have assumed that case is fundamentally a relation between one or more DPs which indicates their relative position in some domain. If this is on the right track, it is reasonable to posit that the mapping from position to case must be MONOTONIC. This would rule out the example in (25), in which the first and third DPs are mapped to nominative, with the second mapped to accusative, much like the *ABA constraint in morphology. Furthermore, this would be just one of several

Phenomenon	One line summary
Spreading	Allow $\text{Nom} \cdot \text{Nom}$, $\text{Acc} \cdot \text{Acc}$, etc.
Dependent case	Grammar can reference the tier sisters of D
Alignment	Does the marked case precede or follow the unmarked case?
Split alignment	$T_{\text{Pres}}/T_{\text{Past}}$ /etc. have different case rules
Case displacement	Lexical case-marked D is invisible on the structural case tier
Diff. argument marking	Grammar can reference non-case features of D

Table 4.1: Summary of case patterns and their TSL-2 analysis

monotonicity requirements affecting case, along with morphological syncretism (Graf 2019), visibility (Bobaljik 2008), and locality (Poole 2022; also see Graf 2020b).

In summary, there may be some unattested case patterns that are TSL-2, but there are far, far more that are not. Once embedded in a larger theory, the limited space of unattested patterns can be further whittled down by the other factors—computational or otherwise—for which we have independent evidence.

4.9 Summary

In this chapter, I have argued that the typology of case dependencies is well explained by the assumption that they are TSL-2. Major points of cross-linguistic variation such as presence of iteration, the order in which cases are assigned, the visibility of both DPs and domain nodes, and interactions between the properties of DPs in the same domain naturally fall out from the parameters of the model. The full list of patterns surveyed in this chapter is summarized in Table 4.1.

Case therefore joins agreement, movement, and long-distance assimilation and dissimilation in exemplifying the close link between the formal typology and computational complexity of long-distance linguistic dependencies. This does not at all mean that the investigation of the subregular complexity of case is complete. As mentioned at the outset, the typological predictions of a multi-tier system of case assignment are not nearly as clear as those for a single tier. The same can be said of movement, agreement, and the interactions among these phenomena. Working this out will require a close examination of the grammars of individual languages, similar to the study of Japanese presented in Chapter 5. We should also carefully investigate the parallels between case and agreement, particularly complex DAM effects, which have been argued by theorists as a reason for unification under the Agree operation (cf. Bárány and Sheehan 2024; Clem and Deal 2024). The present perspective also argues for unification, but of a very different sort: given that case and agreement are built upon the same computations, and given that both can involve multiple DPs which are presumably adjacent on a tier, the existence of such parallels is exactly what we expect.

Chapter 5

Case in Japanese

5.1 Overview

In this chapter, we take our first in-depth look at the grammar of a single language. I develop an analysis of case assignment in Japanese which involves three tiers. The first covers structural case in the verbal domain, the second is for the nominal domain, and the final tier handles lexical case. With a few notable exceptions (Kuno 1973; Ozaki 2014), such a configurational analysis of case in Japanese is rarely attempted in the literature, but as we will see, many detailed facts about the language can be handled with ease in this way.

The chapter is structured as follows. We begin with a preview of the general approach, which at this point should be quite familiar (Section 5.2). Next, I provide an overview of the basic case patterns in Japanese, and propose a set of descriptive generalizations (Section 5.3). Then, I define a set of three TSL-2 grammars which encode these generalizations, then show how the analysis extends easily to more complex structures, particularly those involving cross-clausal case assignment and valency alternations (Section 5.4). I also deal with some interesting formal problems for the tree tier approach which have not yet been discussed, specifically involving adjunction and coordination.

5.2 Preview of the analysis

Consider the example in (1). This is a simple transitive sentence, in which the subject *Taroo* is followed by the nominative marker *ga* and the object *piano* is followed by the accusative marker *o*. A phrase structure tree for this sentence is shown on the left of Figure 5.1. Details of the syntactic analysis will be introduced in Section 5.4. For now, note that the case marker is the head of a K(ase) phrase, and that the subject asymmetrically c-commands the object.

This chapter is based on Hanson (2023) “A TSL Analysis of Japanese Case”. There is one major change, which is that the SS-TSL analysis of the indirect passive and causative constructions in Section 5.4.6 has been replaced with a TSL-2 analysis. I also corrected several small errors. First, I added several citations to Kuno (1973) which were missing in the original. Second, I corrected an error in the definition of the verbal case tier, which allowed a lexical dative as the subject of a non-stative verb. Finally, I corrected a remark in Footnote 6 which suggested that the verbal and genitive case tiers might be combined; this is impossible, as *v* must appear in the former but not the latter.

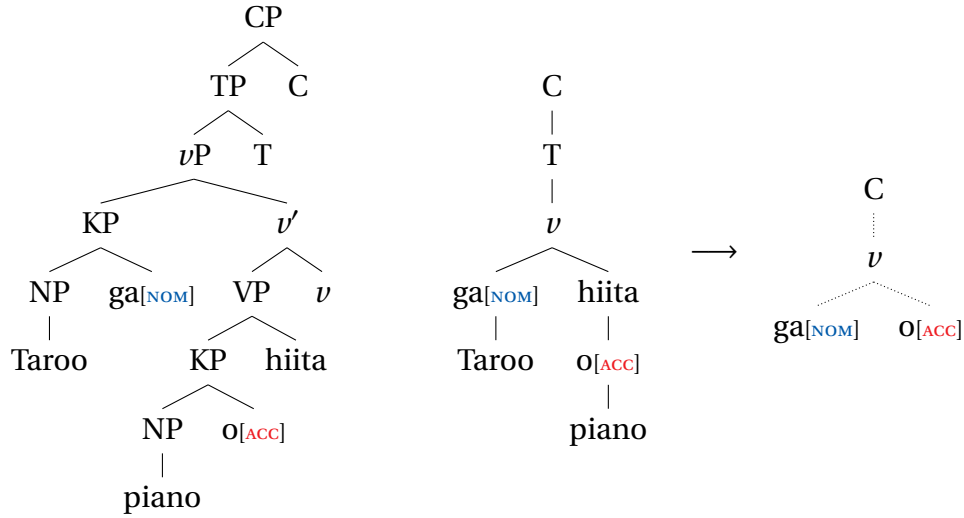


Figure 5.1: Left: phrase structure tree. Right: dependency tree and tier projection enforcing accusative constraint. Nodes of category K, C, and ν are projected. It is required that every o has a ga among its left sisters.

- (1) Taroo $ga/*o$ piano $o/*ga$ hiita.
 Taroo NOM/ACC piano ACC/NOM played

‘Taroo played the piano.’

On the right of Figure 5.1 is the dependency tree corresponding to the phrase structure tree, along with the case tier projection. Each node in the dependency tree is a lexical item, taking the place of a head and all of its projections in the phrase structure tree. The ν node has two daughters, corresponding to its specifier (the subject, headed by the case marker) and its complement (VP, headed by the verb). Other nodes have only a single child, corresponding to their complements. A full dependency tree would display the features of each node; for brevity I omit everything but the node label and relevant features such as case, using the category as the label of empty elements such as C/T/ ν .

This brings us to the tier projection. The general approach taken in this chapter will be to construct a tier such that all nominals in some case licensing domain become daughters of the domain node, and to state the constraints on case configurations over the daughter strings of the domain nodes. In the present example, the relevant constraint (simplified) is that the accusative marker o must be c-commanded by nominative ga in the same clause. To enforce this constraint, we project a tier which includes all nodes of category ν , K, and C. Since dominance and precedence are preserved, ga and o become daughters of ν . We then require that ga be a left sister of o .¹ The TSL grammar for the daughter string language of ν will thus ban substrings such as $o\ ga$. The full analysis, which contains many additional constraints over several tiers, will be fleshed out in Section 5.4.

¹In principle it is possible for a left sister on the tier not to be c-commander in the dependency tree. In practice this turns out not to be an issue. See Section 5.4.7 for an example.

It is worth noting that in an MG dependency tree all elements appear in the position of first merge—when a nominal has moved, such as by passivization or scrambling, only its base position is considered for purposes of case licensing. This assumption has been adequate for all phenomena examined in this framework to date, and it also seems to be appropriate for case in Japanese. Scrambling, for example, is widely understood to preserve case marking. Even when a certain case correlates with movement (as in some analyses of differential object marking), it is usually possible to predict the case of nominal based on its context and its other features (e.g. definiteness). Since this issue does not arise in the Japanese data, I say no more here.

5.3 Descriptive generalizations

Japanese has four core cases, marked by the suffixes *ga* (nominative), *o* (accusative), *ni* (dative), and *no* (genitive). Their prototypical functions are similar to German and other Indo-European languages: subjects receive nominative case, direct objects receive accusative, and indirect objects receive dative, while complements and possessors of nouns are genitive. Examples of simple intransitive (2a), transitive (2b), and ditransitive sentences (2c) are given below, along with examples of a nominal complement (2d) and possessor (2e). All examples are presented in topic-less sentences since topic marking masks the underlying case.²

- (2) a. Taroo *ga* hasitta.
Taroo **NOM** ran
'Taroo ran.'
- b. Taroo *ga* piano *o* hiita. (=1)
Taroo **NOM** piano **ACC** played
'Taroo played the piano.'
- c. Jin *ga* Yumi *ni* hon *o* ageta.
Jin **NOM** Yumi **DAT** book **ACC** gave
'Jin gave Yumi a book.'
- d. Taroo *ga* [yama no e] o mita.
Taroo **NOM** mountain **GEN** picture **ACC** saw
'Taroo saw a picture of a mountain.'
- e. Taroo *no* hon
Taroo **GEN** book
'Taroo's book'

In general, nominative, accusative, and dative case are available for arguments of verbs, and the number of arguments predicts what their cases should be: if there is one argument then it is nominative; if there are two then the latter is accusative, and if there are more than two then the middle nominals are dative. This is also true for complex verbal predicates, with

²Data is adapted from Miyagawa (1989) unless noted otherwise.

some complications (discussed in Sections 5.4.5 and 5.4.6). Conversely, arguments of nouns are usually genitive no matter how many there are. An example of a noun phrase multiple genitive arguments is given in (3) below.

- (3) Taroo no yama no e
 Taroo GEN mountain GEN picture
 ‘Taroo’s picture of a mountain’

While these are the canonical patterns, several others are possible. Some transitive verbs take a dative object rather than the usual accusative (4a). Additionally, stative verbs such as *dekiru* ‘can do’ take a nominative object, and allow dative and/or nominative for the subject (this varies depending on the exact verb), yielding dative-nominative (4b) and double nominative (4c) structures. Transitive adjectives and complex verbs formed with a stative suffix also allow nominative objects (cf. Kuno 1973).

- (4) a. Taroo ga Yumi ni atta.
 Taroo NOM Yumi DAT met
 ‘Taroo met Yumi.’
 b. Yumi ni tenisu ga dekiru.
 Yumi DAT tennis NOM can.do
 ‘Yumi can play tennis.’
 c. Yumi ga tenisu ga dekiru.
 Yumi NOM tennis NOM can.do
 ‘Yumi can play tennis.’

Of the four cases, nominative has the widest distribution. As we will see later (Sections 5.4.4 and 5.4.6), it can also be replaced with another case when a verb or adjective and its arguments are embedded in a larger structure. Thus, it makes sense to treat nominative as the *default* case, appearing when no other condition applies.

To briefly summarize, the case that marks a nominal in some domain depends primarily on (1) the category of the domain and (2) the position of that nominal relative to others in the domain. Additionally, certain predicates specify that one of their arguments must be dative rather than the case that would otherwise be expected. Specifically, we could say that accusative and genitive are *structural* cases (i.e. licensed by the structural context); some instances of dative are structural while others are *lexical* (licensed by specific lexical items); finally, nominative is the *default*.

I am not aware of any work in the syntactic literature that analyzes the entire case system of Japanese in this manner. However, the individual patterns are well-known, and much of the analysis is a direct application of ideas from dependent case theory (Marantz 2000; Baker 2015), supplemented with insights by Kuno (1973). The primary purpose of this chapter is to show that the generalizations outlined above are easily implemented using a TSL grammar, and that they can be extended to more complex constructions with little or no modification. As a computational analysis, it is essentially descriptive in nature, and in principle compatible

with a variety of theories of case licensing. At the same time, most of the patterns discussed here (or close analogues) can also be found in other rich case-marking languages, so there is good reason to believe that the approach should generalize beyond Japanese.

5.4 The analysis

We are almost ready to formalize the generalizations made in the previous section. Before we begin, I lay out a few syntactic assumptions. First, clauses are assumed to have the following functional hierarchy:

$$C > T > (\text{Pass}) > (\text{Caus}) > \nu > (\text{Appl}) > V$$

In essence, this is a modern version of the “biclausal” analysis for the passive and causative constructions. These heads may be considered subtypes of ν , labeled separately for convenience. Next, goals of ditransitive verbs may appear in two positions: low goals are PP daughters of VP, while high goals are KP daughters of an Appl(icative) head (Miyagawa and Tsujioka 2004). This fact will be relevant to the analysis of passivization in Section 5.4.5. Finally, nominals are treated as NPs, with case markers occupying a higher KP.³

The remainder of this section is structured as follows. First, I introduce three tree tiers corresponding to structural cases licensed in the verbal domain, the nominal domain, and lexical case. Next, I consider more complex constructions, including embedded clauses, passives, and causatives, making several small revisions. From there, I refine the analysis to handle adjuncts, and address a potential problem involving coordination.

5.4.1 Accusative and structural dative case

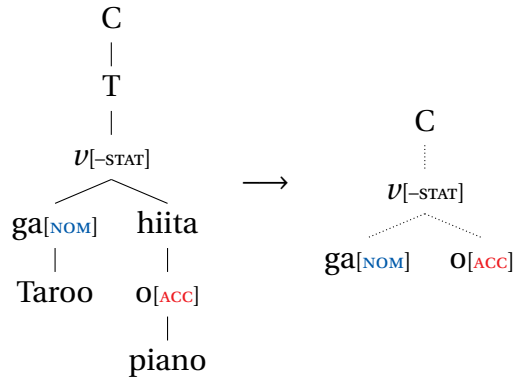
First, we define a tier to license structural cases in the verbal domain: accusative and dative. On this tier we project non-stative ν and all K heads. We also project C heads in order to limit the case licensing domain to a single clause; while ν in the embedded clause will normally do this, it will not always be present on the tier.

The constraints on the tier are, roughly: (1) the rightmost of two or more K children of ν must bear accusative, (2) the middle of three or more K heads must bear dative, and (3) no other K heads may bear accusative or dative. Other K heads are underspecified; if not specified as genitive or lexical dative on the relevant tiers they become nominative by default. This includes all subjects as well as objects of stative verbs (since stative ν is not projected).

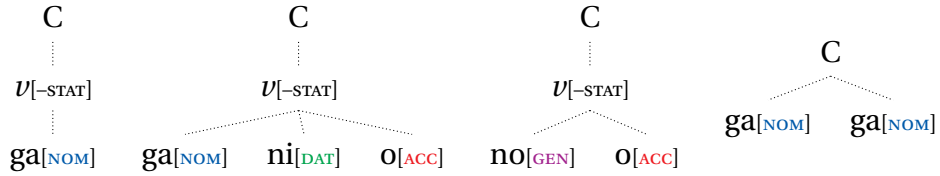
An example for the simple transitive sentence in (1) is given in Figure 5.2a. Since non-stative ν is projected, the tier is unchanged from the example in Section 5.2. The daughter string of ν satisfies the constraints just mentioned: the accusative K is the rightmost of two K children of ν and the nominative K meets the elsewhere criterion. Further examples of well/ill-formed tiers are given in Figures 5.2b and 5.2c, respectively.⁴

³PPs take an NP complement instead of a KP. PPs may alternatively be analyzed as KPs bearing semantic case. Such cases do not need to be licensed syntactically.

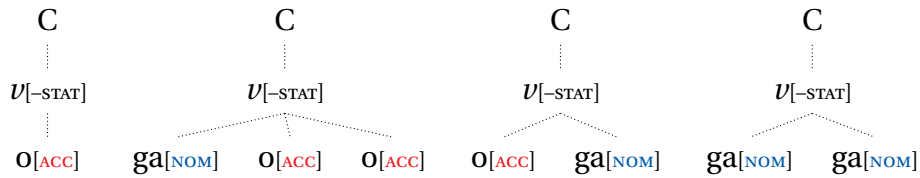
⁴The feature $[\pm\text{STAT}]$ is not an MG feature. It is nonetheless placed in brackets for readability.



(a) Dependency tree and verbal case tier for a transitive sentence. The subject licenses the accusative object to its right.



(b) Other licit verbal case tiers. From left to right: intransitive verb, ditransitive verb, genitive subject, stative verb.



(c) Examples of illicit verbal case tiers. In order: lone accusative, double accusative, accusative before nominative, non-stative double nominative.

Figure 5.2: Examples of licit and illicit verbal case tiers

This largely recapitulates the analysis in Kuno (1973). To my knowledge, no work since his has treated Japanese structural cases in precisely this manner. In particular, the treatment of structural dative as a kind of “doubly dependent case”, requiring the presence of a KP both before and after in the same clause, appears to be unique to his work. And yet, it is extremely natural in the present system, and unlike the treatment in Baker (2015), requires no reference to phases.

Moving along, we must also take into account the fact that lexical datives are allowed as direct objects. It turns out that many verbs are compatible with either an accusative or dative object, with a difference in temporal properties (cf. Fukuda 2007); I assume such verbs to be optional licensors of lexical dative case. The full tier definition is given below.⁵

(5) Verbal case tier (initial version)

Project categories: $\{v_{[-\text{STAT}]}, K, C\}$

Daughter string languages:

v : $\{\text{NOM}, \text{GEN}\} \cdot (\text{DAT}^* \cdot \{\text{ACC}, \text{LDAT}\})$

K/C : $\overline{\{\text{ACC}, \text{DAT}\}}^*$

For clarity, all daughter string languages are defined using regular expressions. Since it may not be immediately obvious that these languages are TSL (or SL), I also provide grammars for the verbal tier:

- The daughter string language of v is SL-2. Its grammar is:
 $G^- = \{\$, \text{DAT}, \$ \cdot \text{ACC}, \text{NOM} \cdot \text{NOM}, \text{NOM} \cdot \text{GEN}, \text{GEN} \cdot \text{NOM}, \text{GEN} \cdot \text{GEN}, \text{DAT} \cdot \$, \text{DAT} \cdot \text{NOM}, \text{DAT} \cdot \text{GEN}, \text{ACC} \cdot \text{NOM}, \text{ACC} \cdot \text{GEN}, \text{ACC} \cdot \text{DAT}, \text{ACC} \cdot \text{ACC}, \text{ACC} \cdot \text{LDAT}, \text{LDAT} \cdot \text{NOM}, \text{LDAT} \cdot \text{GEN}, \text{LDAT} \cdot \text{LDAT}\}$
- The daughter string language of K/C is SL-1. Its grammar is:
 $G^- = \{\text{ACC}, \text{DAT}\}$

Small modifications to the verbal case tier will be required; the revised tier definition is given in Section 5.4.6. Also, while the current daughter string languages are SL, they will later be converted to TSL to accommodate adjuncts (Section 5.4.7).

5.4.2 Genitives

Next, we turn to genitives, which have the simplest distribution: as a first approximation, all KPs in the domain of a nominal are genitive, and no others. We construct the genitive tier as follows:

(6) Genitive case tier (initial version)

Project categories: $\{N, K, C\}$

⁵String languages are notated using regular expressions. $\text{NOM}/\text{ACC}/\text{etc.}$ stand for a K head bearing said case. A dot ($\cdot$) represents concatenation. Parentheses mean *optional*. Set braces represent a choice among alternatives. An overbar represents set complement.

Daughter string languages:

N: $\{\overline{\text{GEN}}, \text{N}, \text{C}\}^*$

K/C: $\{\overline{\text{GEN}}\}^*$

The tier projection for (2d), in which the object nominal contains a genitive complement, is shown in Figure 5.3a). There is only a single K child, the noun *e* ‘picture’, and it bears **GEN** as required, so the tier is well-formed. There are no restrictions on the other KPs, though they could of course be ruled out on other tiers.

Subjects of certain embedded clauses appear in genitive case, an apparent exception to the current tier definition; this will require modification as discussed in Section 5.4.4. Another issue worth noting is that the particle *no* can appear between PPs and their head nouns, as in example (7) below. This *no* is traditionally considered to be a marker of adnominal modification rather than a case particle. Fortunately, we can abstract away from this issue. If these instances of *no* are case particles, then they still adhere to the constraint as stated; if not, then the constraint simply does not apply.

- (7) *otera e no michi*
 temple to NO road
 ‘the road to the temple’

5.4.3 Lexical datives

The third case tier controls the distribution of lexically dative-marked nominals. While we could reasonably leave lexical case to be handled by the selection (i.e. subcategorization) mechanism, it is worth demonstrating that both structural and lexical case can be regulated in a unified manner if desired. Lexical dative may be assigned to an argument of either *v* or V depending on the verb, but only *v* appears on the verbal case tier, so a new tier is required which projects both. On this tier, verbal heads licensing a lexical dative KP must have exactly one such daughter; lexical dative KPs may appear nowhere else. The tier is defined as follows:

(8) **Lexical case tier**

Project categories: $\{v, V, K, C\}$

Daughter string languages:

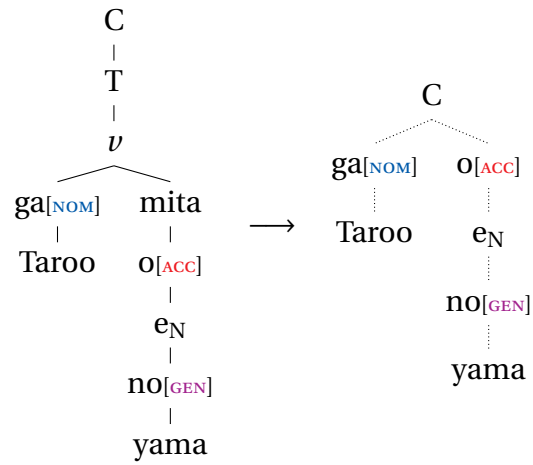
v/V (**LDAT** licenser): $\{\overline{\text{LDAT}}\}^* \cdot \text{LDAT} \cdot \{\overline{\text{LDAT}}\}^*$

v/V (non-**LDAT** licenser): $\{\overline{\text{LDAT}}\}^*$

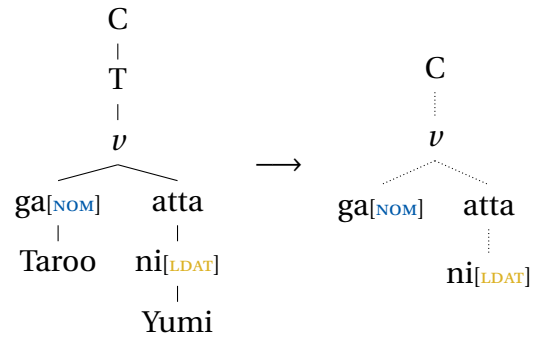
K/C: $\{\overline{\text{LDAT}}\}^*$

The tier projection for (4a), in which the verb requires a lexical dative object, is given in Figure 5.3b. The only KP child of V is a lexical dative, so the tier is well-formed. If the dative licenser was *v* then the subject would need to be dative instead.

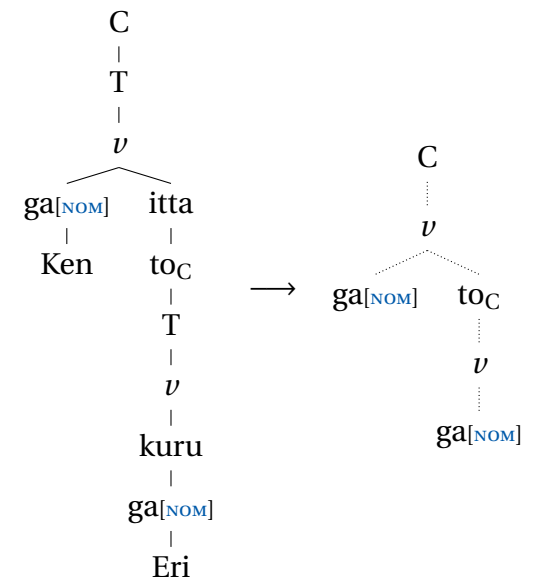
While it might be desirable to use the same feature for both structural and lexical datives, this would prevent structural datives from being ruled out in subject and direct object position. Since lexical datives differ in behavior from other nominals (they cannot be passivized



(a) Genitive tier for nominal complement



(b) Lexical case tier for dative object verb



(c) Verbal case tier for embedded CP

Figure 5.3: Example tiers for genitive, lexical dative, and an embedded clause

in Japanese, for example), such a distinction seems appropriate. In effect, we are treating structural and lexical dative as different cases.

One question that this analysis raises is what should happen if there are two KP children of the same lexical dative case licenser, in particular V. As far as I am aware, this situation never arises in Japanese. If it does in other languages, then the grammar must specify which KP should be dative.

Now, having defined three tiers modeling the canonical uses of the four core cases, we will consider more complex structures, and see that the system can handle them with minimal adjustment.⁶

5.4.4 Embedded clauses

There are several types of finite embedded clauses in Japanese. By default, these show the same case marking as matrix clauses, but under certain circumstances the embedded subject may be marked accusative or genitive.

We will first confirm that the analysis works correctly for the basic pattern. Examples of two types of finite embedded clauses are given in (9) below. Here, *to* is analyzed as a complementizer, while *koto* is a noun taking a CP complement.

- (9) a. Ken ga [Eri ga kuru to] itta.
 Ken NOM Eri NOM come C said
 ‘Ken said that Eri will come.’
 b. Eri ga [Ken ga tegami o okutta ∅] koto o sitteiru.
 Eri NOM Ken NOM tegami ACC sent C thing ACC know
 ‘Eri knows that Ken sent the letter.’

Since we project C on the tier, a new case domain is created for each embedded clause, resulting in the same case configuration as in a matrix clause. As an example, the tier projection for sentence (9a) is shown in Figure 5.3c). While projecting C may seem redundant, it is necessary because *v* is not projected on the verbal tier when it is stative. It also provides the basis for the analysis of the alternative case marking patterns, which we now turn to.

In the Japanese ECM construction, the embedded subject appears to take accusative case (10). If this nominal was a matrix object binding a *pro* subject in the embedded clause (a prolepsis analysis) then there would be nothing to explain, but Kishimoto (2018) argues that at least some ECM subjects are genuine. Similarly, in GA-NO CONVERSION the subject of a nominal clause takes genitive case (11). Both structures are also grammatical with a nominative embedded subject.

⁶As noted by a reviewer of Hanson (2023), it has been suggested for phonology that when a dependency involves multiple tiers, the tier alphabets are either nested or disjoint, but never incomparable (Aksénova and Deshmukh 2018); this generalization has since been challenged (Gleim and Schneider 2023). Since lexical dative is always assigned locally the tier alphabet could be expanded to all categories (in effect, an SL tree grammar), making it a superset of the others. It is not clear whether this can be achieved for the verbal and genitive tiers. Furthermore, when we look at the whole syntactic grammar (not just case licensing), partially overlapping tiers are unavoidable.

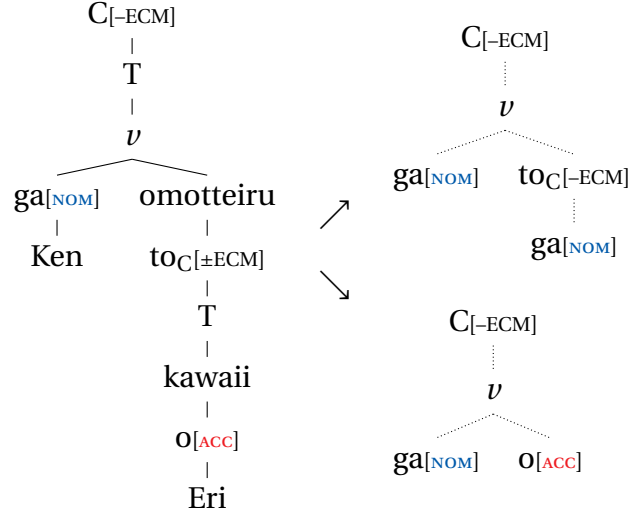


Figure 5.4: Verbal case tier for embedded clause with and without ECM. C_{+ECM} is not projected, so the embedded subject becomes part of the higher case domain.

(10) Finite ECM (Kishimoto 2018)

Ken ga [Eri {ga/o} kawaii to] omotteiru.

Ken **NOM** Eri {**NOM/ACC**} be.cute C think

‘Ken thinks that Eri is cute.’

(11) Ga-no conversion (Maki and Uchibori 2008)

Eri ga [Ken {ga/no} kita ∅] riyuu o sitteiru.

Eri **NOM** Ken {**NOM/GEN**} came C reason **ACC** know

‘Eri knows the reason that Ken came.’

This variable cross-clausal case marking may seem mysterious, but in fact all that is needed to derive the patterns is to selectively ignore the embedded C head. The easiest way to do this is to posit that the relevant lexical predicates may select a C head with a special feature, call it [+ECM].⁷ Such C heads are not projected, similar to our treatment of stative verbs. This approach also has precedent in work which attributes variation in cross-clausal dependencies to the feature composition of the complementizer (cf. Lohninger et al. 2022). Indeed, Hiraiwa (2001b) claims that ga-no conversion involves a special complementizer.

Example tier projections for (10) are shown in Figure 5.4 (the treatment of (11) is exactly parallel). Revised tier definitions are provided in Section 5.4.6. For simplicity, I treat the subject of an adjective as its complement, and ignore the aspectual morphology of *omotteiru*. All said, the facts about embedded clauses are handled quite well under the TSL perspective.

⁷Like [±STAT], this is not an MG feature.

5.4.5 Passives

Next, we examine complex predicates within a single clause, formed with the passive suffix *-rare* and the causative suffix *-sase*. The passive suffix itself has at least two functions: the DIRECT PASSIVE, which decreases the valency of a transitive verb by eliminating the agent, and the INDIRECT PASSIVE, which increases valency. The literature disagrees on exactly how many distinct lexical items exist (see Ishizuka 2017 for an overview). I assume two homophonous passive suffixes corresponding to the two major functions. Recall also that I assume these suffixes to realize distinct functional heads, though the analysis could also work with verbs bearing ‘passive’ and ‘causative’ features.

The direct passive will be the focus on this section; the indirect passive will be discussed together with causatives. An example is given in (12).

(12) Active/passive transitive verb

- a. Sensei ga gakusei o hometa.
teacher **NOM** student **ACC** praised
‘The teacher praised the student.’
- b. Gakusei ga (sensei ni) homerareta.
student **NOM** teacher by praised.PASS
‘The student was praised (by the teacher).’

The object of a passivized transitive verb is promoted to the subject, and receives nominative case. These facts are straightforwardly understood under the common assumption that agent is not projected in Spec-*ν*P in passives, and that the optional *by*-phrase is an adjunct PP. Miyagawa (1989) argues that this is indeed the case in Japanese.

For ditransitive verbs, there are two possibilities: either the (higher) goal is promoted, or the (lower) theme is promoted. Example (13) shows an active ditransitive verb along with the corresponding goal (13b) and theme (13c) passives (optional *by*-phrases are omitted).

(13) Active/passive ditransitive verb

- a. Mari ga kodomo ni okasi o ataeta.
Mari **NOM** child **DAT** candy **ACC** gave
‘Mari gave the child candy.’
- b. Kodomo ga okasi o ataerareta.
child **NOM** candy **ACC** gave.PASS
‘The child was given candy.’
- c. Okasi ga kodomo ni ataerareta.
candy **NOM** child **DAT** gave.PASS
‘The candy was given to the child.’

The goal passive (13b) requires no special explanation assuming that dative case here is structural. Once the agent is eliminated, there are only two arguments, effectively creating

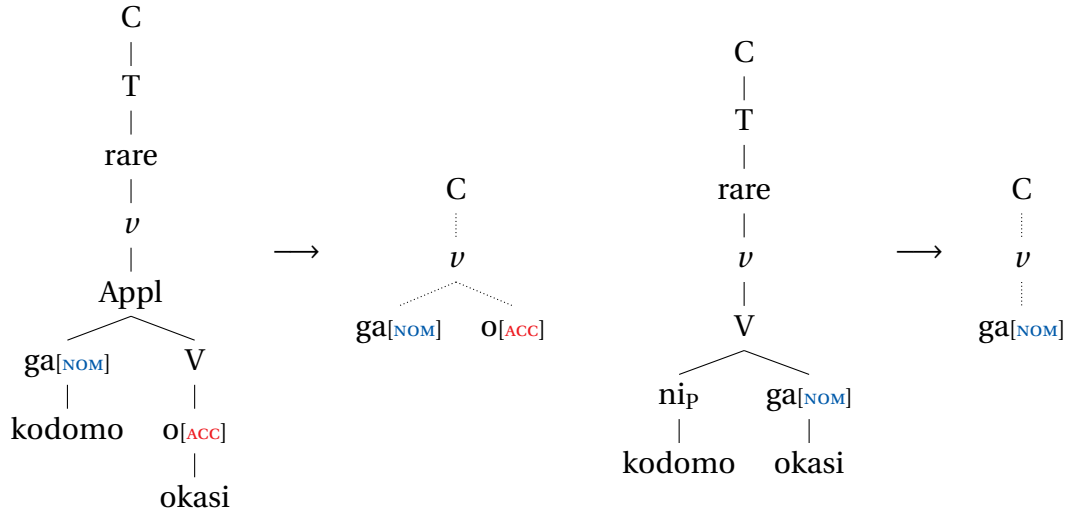


Figure 5.5: Verbal case tiers for passives of ditransitive. Left: goal passive (goal is a KP). Right: theme passive (goal is a PP).

a transitive verb. There is an elegant solution for the theme passive as well. Recall that the goal of a ditransitive verb may occupy one of two positions, and that the higher position is a KP while the lower position is a PP. Thus, it should be possible to target the direct object for promotion by selecting the low goal structure. So we see that the facts about passives fall out naturally in this analysis. Dependency trees and verbal case tiers for the ditransitive goal passive and theme passive are shown in Figure 5.5.

5.4.6 Causatives

As our final case study, we consider the causative construction. The causative morpheme *sase* is compatible with verbs of any valency. Causative equivalents of the examples in (2) are given in (14) below. The causee of an intransitive verb may be accusative or dative, corresponding to the *make* or *let* interpretations, respectively. For other verbs the causee must be dative, and either interpretation is possible. I set aside these semantic details.

- (14) a. Ken ga Taroo {ni/o} hasiraseta.
 Ken NOM Taroo DAT/ACC ran.CAUS
 ‘Ken made/let Taroo run.’
 b. Ken ga Taroo ni piano o hikaseta.
 Ken NOM Taroo DAT piano ACC played.CAUS
 ‘Ken made/let Taroo play the piano.’
 c. Ken ga Jin ni Yumi ni hon o agesaseta.
 Ken NOM Jin DAT Yumi DAT book ACC gave.CAUS
 ‘Ken made/let Jin give Yumi a book.’

In an intransitive sentence the causee may be dative without an accompanying accusative object, suggesting lexical dative case. Additional arguments appear in the expected cases,

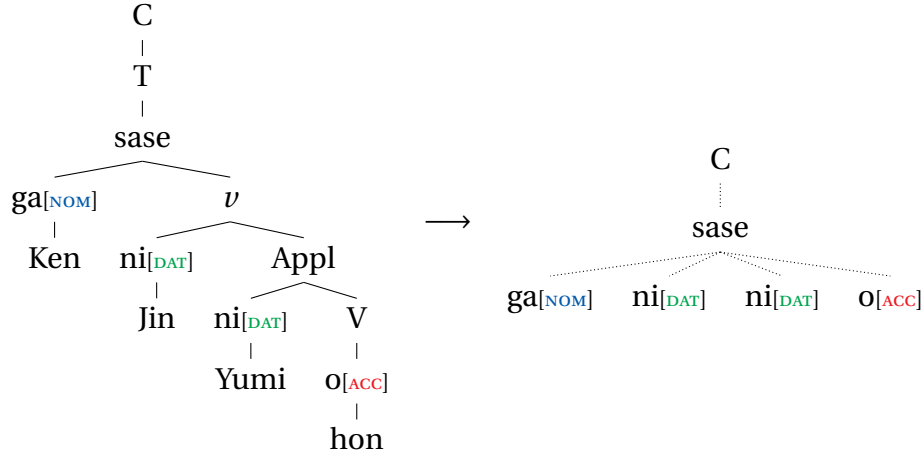


Figure 5.6: SS-TSL analysis of the causative of a ditransitive verb. All verbal arguments are daughters of *sase*, allowing the causee to be assigned dative in the usual manner.

suggesting that these are the usual structural cases. But in the present system it is the causer that would receive dative case from *sase*, not the causee. Furthermore, it is possible to passivize a causative, in which case the causee becomes nominative like any other structurally case-marked nominal, as shown in (15).

- (15) Taroo ga (Ken ni) hasiraserareta.
 Taroo **NOM** Ken by run.CAUS.PASS.PST
 ‘Taroo was made to run (by Ken).’

There are several possible solutions. One is to add additional case tiers corresponding to each functional head, allowing each to restrict the case of the second K child of that head. So, a new *causative tier* would determine the case of causee, leaving the case of the causer up to the next higher tier. While this appears to be technically feasible, we would ideally like to treat simplex and derived verbal structures in a unified manner, since they display nearly the same case patterns.

In Hanson (2023), I proposed a solution which utilizes structure-sensitive tier projection to construct the tree tier, as in Graf (2018) and Vu et al. (2019). At that time, I was not particularly focused on the TSL-2 hypothesis, but rather, how to achieve the most elegant analysis. For completeness, I present the original version first, then address how we might achieve a similar outcome using only TSL-2.

For the SS-TSL analysis, we project the *highest* *v* head in each clause, that is, the one selected by T, increasing the context to a window of height 2. Then, we let the identity of the *v* head determine its daughter string language. This analysis is depicted in Figure 5.6.

The indirect passive (adversative passive) can also be handled in the same manner. Examples of this construction are given in (16) below.

- (16) a. Ken ga Taroo ni hasirareta.
 Ken **NOM** Taroo **DAT** run.PASS.PST

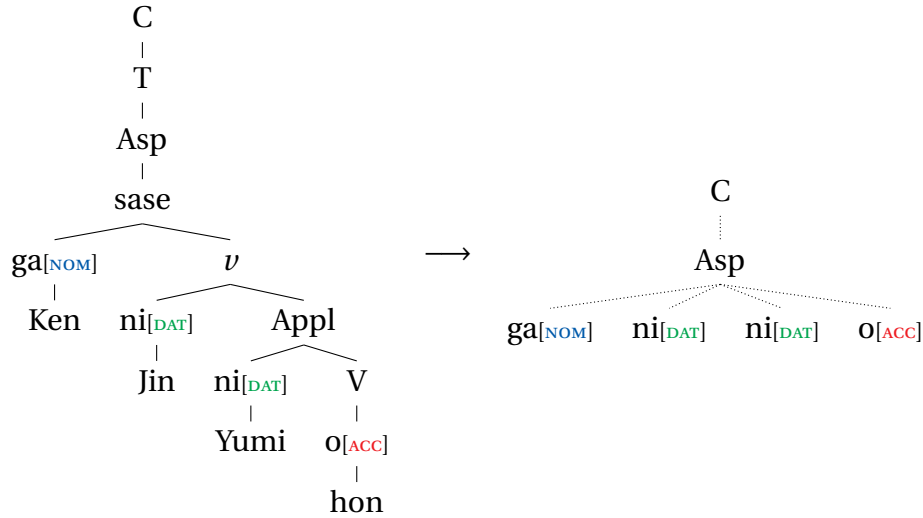


Figure 5.7: TSL-2 analysis of the causative of a ditransitive verb. The Asp head is now the sole assigner of accusative and structural dative, eliminating the need for structure-sensitive tier projection.

‘Ken was annoyed by Taroo running.’

- b. Ken ga Taroo ni piano o hikareta.
 Ken NOM Taroo DAT piano ACC play.PASS.PST

‘Ken was annoyed by Taroo playing the piano.’

Unlike in the causative construction, the embedded subject always receives dative case, regardless of position. Since the highest verbal morpheme determines the daughter string language, this is not a problem.

As far as I can tell, this analysis works. However, not only does it require structure-sensitive tier projection, it is no longer the case that all daughter string languages are SL-2. This is because the causee/affectee may be dative in final position (optionally for causatives, obligatorily for indirect passives), but only when there are two arguments total. Ensuring this requires a window of at least size three. It may be possible to restore the window to size two if we assume that these exceptional datives are actually instances of lexical dative. This requires a reworking of the lexical case tier (more on this below).

Now, let us consider whether we can eliminate structure-sensitive tier projection. Recall that we are assuming that non-stative ν assigns accusative and structural dative, while stative ν is completely inert for purposes of case assignment. Since stativity is an aspectual distinction, it is not unreasonable to posit an Asp(ect) head which occurs below T but above all of the verbal morphemes. We could then relocate assignment of accusative and dative to the non-stative version of this head, while the stative version is either invisible or simply not present. The result is a tier which is identical to that of the SS-TSL analysis except that the Asp head appears in place of the highest verbal head. This is illustrated in Figure 5.7.

Incorporating the other modifications from the last several sections, we arrive at the final definitions for the verbal and genitive case tiers, given in (17).

(17) **Verbal case tier (revised)**

Project categories: $\{\text{Asp}_{[-\text{STAT}]}, \text{K}, \text{C}_{[-\text{ECM}]}\}$

Daughter string languages:

Asp: $\{\text{NOM}, \text{GEN}\} \cdot (\{\text{DAT}, \text{LDAT}\}^* \cdot \{\text{ACC}, \text{LDAT}\})$

K/C: $\{\text{ACC}, \text{DAT}\}^*$

(18) **Genitive case tier (revised)**

Project categories: $\{\text{N}, \text{K}, \text{C}_{[-\text{ECM}]}\}$

Daughter string languages:

N: $\{\text{GEN}, \text{N}, \text{C}\}^*$

K/C: $\{\text{GEN}\}^*$

As with the SS-TSL analysis, this analysis is slightly incomplete, as it relies on lexical dative being assigned correctly in both intermediate and final argument positions. This on its own is not a problem. However, as I noted earlier in this section, the causative and indirect passive morphemes appear to assign lexical dative case at a short distance, skipping the argument they introduce in favor of the next one below. This is quite different from the assignment lexical dative assigned by ν/V as treated in Section 5.4.3. In this case, the best option might be to reconsider the structural analysis: if the passive and indirect passive involve control structures, then the causee/affectee is in fact local to the case-assigning head. It would also be prudent to confirm whether the Aspect head posited here has independent support. I leave a more thorough investigation of these issues to future work.

5.4.7 Adjuncts

Adjuncts such as adverbs and PPs interfere with the tier constraints as currently defined, since there is in principle no bound to the number that may appear in any given case domain. We would like to ignore them since their presence does not affect case licensing. However, we cannot omit them on the tree tiers because any K heads they dominate would be interspersed among the daughters of a higher head. Instead, we must modify the daughter string languages, converting them from SL to TSL languages and ignoring adjuncts at this stage.

To see why this works, recall again the form of the daughter string language of ν on the verbal case tier, which (simplified) has the form $a(b^*c)$. Representing adjuncts by the symbol x and allowing them to occur anywhere, our string language is now $x^*a((x^*b)^*x^*c)x^*$. If we project a tier omitting x then the tier language is once again $a(b^*c)$.

When originally devising the above analysis, the approach I had in mind was to use CATEGORY-PRESERVING SELECTION by means of adjunctizer heads. For example, the adjunctizer head for adjectives selects for categories A and N and itself is an N. The number of such heads in any MG lexicon is finite. We add all such heads to all *tree* tier projection functions, but omit them from the daughter string tier projection function as just described. It is also possible to introduce a separate adjunction operation into the grammar, as in Frey and Gärtner (2002), without disrupting the complexity of other operations; see Hanson (2025a)

for details. In the latter approach, we pick out adjuncts as exactly those nodes bearing an adjunction feature. This is a small theoretical improvement, as it allows us to maintain the generalization that the tier alphabet targets entire classes of elements as encoding by other aspects of the grammar formalism.

5.4.8 Coordination

Due to the complexity of the data and the number of theories of coordination, it is beyond the scope of this chapter to consider these in any depth, but I wish to at least outline the general problem and what it means for the analysis. Essentially, coordination is a problem when it splits a case domain, such as when a ν P contains a coordinated VP. When we project the verbal case tier, the children of both Vs all end up as daughters of ν ; this predicts a change in case marking, which is contrary to fact.

Does this issue actually arise in Japanese? Perhaps not. Japanese allows coordination of TP and ν P but not VP, and subjects may optionally remain in situ (cf. Hirata 2006, and references therein). This means that whether each verb phrase has its own subject (remaining in situ), or both share a single subject (raised via across-the-board movement), there is no conflict. In a language similar where VP coordination is possible, we would need to restructure the analysis to include additional nested case domains (we avoided this earlier for the passive and causative constructions by using structure-sensitive tier projection). Should this prove unfeasible, this seems to be the most likely way in which the tree tier-based analysis could be invalidated.

It is at this point that I should note an alternative generalization of TSL to trees based on so-called *c[ommand]-strings* which, roughly speaking, encode chains of c-commanding elements (Graf and Shafiei 2019). Because the present analysis already operates by collecting nominals in the daughter string of the case licensing domain node, it should be straightforward to recast it in terms of c-strings. This new version would also be robust against the domain-splitting problem presented by coordination. I leave the investigation of this possibility to future work.

5.5 Summary

In this chapter, I developed a TSL analysis of Japanese case, and showed that the descriptive generalizations are captured by a system of three tiers and a small number of constraints, and that the analysis extends with minimal modification to a wide range of constructions. The analysis is simple in computational terms and concise as a description of the case patterns of Japanese. These results support the proposal that the syntactic distribution of case is TSL. As the reader has no doubt noticed, many of the case patterns discussed in this chapter also have close parallels in other languages. It seems likely that, once additional in-depth case studies of this kind are conducted, we will be able to learn even more about the comparative facts of case-rich languages.

Chapter 6

Modeling Agreement

6.1 Overview

In Chapter 2, I introduced two approaches to generalizing TSL-2 string constraints to trees for use in syntax. The first was the tree tier model, originally developed for movement, and applied here to case assignment in the previous three chapters. The second was the command string model, originally proposed by Graf and Shafiei (2019) to model licensing dependencies. This will be starting point for the model of agreement presented in this chapter.

This time, I have less to say about the theoretical background, as I will adopt the standard perspective of contemporary Minimalism starting with Chomsky (2000), which treats agreement as involving matching pairs of features: an UNVALUED feature called a “probe” searches the tree for a matching VALUED feature, called the “goal”. Instead, I focus on a few additional formal aspects are needed to fully operationalize the TSL-2 model of agreement. First, how do we represent agreement in an MG dependency tree? Second, how do we describe the set of licit agreement configurations using TSL-2 grammars over command strings?

For the representation, I adapt the binary feature notation already in use for movement to agreement, using positive features to represent “probes” and negative features for “goals”. Regarding the second question, I define a variant of a command string which I call a SPINE. This will allow us to define our grammars in an extremely straightforward manner, which is almost exactly parallel to the treatment of long-distance phonotactic dependencies as discussed in Chapter 2. As before, this model provides the foundation for the typological study in Chapter 7 and the in-depth analysis of Hindi in Chapter 8.

The rest of this chapter proceeds as follows. I begin with the scope of investigation and comments on a few relevant theoretical issues (Section 6.2). Next, I introduce the complement spine model and how it compares to ordinary command strings (Section 6.3). Finally, I demonstrate how the system can be used to model agreement, using examples from English (Section 6.4). As with the model of case introduced in Chapter 3, this grammar is intended mainly for illustrative purposes. In the following chapters, we will explore the consequences of the system in much greater depth.

This chapter is based on material from Hanson (2025c) “Tier-based strict locality and the typology of agreement”. Section 6.2 is the main substantive addition. I also updated the format of the TSL-2 grammars to use two variables X and Y for the two elements in each constraint.

6.2 What is agreement?

Like case, agreement has both a syntactic and a morphological component. Following standard assumptions, certain nodes include information on person, number, gender or noun class, and perhaps other features. When two nodes agree, they must match in some or all of these features. This may be realized in the morphology with inflectional affixes or suppletive forms, which may be fully transparent or partly fused, perhaps with syncretism among certain feature combinations, with many interesting complexities besides (see Corbett (2006) for an overview). We are exclusively interested in the syntactic side, which boils down to identifying the items in the tree which enter into an agreement relation, so I say no more about the morphology here.

Our attention will be on agreement between verbs (and other clausal elements) and DPs. Traditionally, generative syntacticians have held that the finite T head (formerly Infl) is responsible for subject-verb agreement in many languages, as evidenced by the fact agreement tends to disappear in non-finite contexts. Both tense and agreement information are then attached to the finite verb in a separate step, perhaps via head movement of V to T, as in many Indo-European languages, or by “affix hopping” onto the verb in the case of English. In agglutinative languages, no further operations may be necessary to get the correct morpheme order, though head movement is often assumed nonetheless. In this work, I focus on the initial step which gets the features of the subject onto T, though a version the second step is included in the analysis of Hindi in Chapter 8.

Other categories in the clausal spine may also agree, either with the subject or another DP. The locus of object agreement (in languages that have it) is often thought to be the *v* head. Beyond this, many languages have complementizer agreement of some form, and the head noun of the DP may agree with the possessor. Although I do not directly treat object agreement or possessor agreement, I do include several cases of complementizer agreement in the cross-linguistic study in Chapter 7.

If we look further, even more types of agreement can be distinguished. In particular, many languages display what is often called CONCORD among the various constituents of the DP, duplicating some subset of person/number/gender on multiple heads in various combinations. From the present perspective, this can be considered a type of agreement, as it involves feature matching among nodes. The major distinction seems to be that while agreement proper is best thought of as a pairwise relation between a DP and some other head, concord involves spreading of features to any relevant heads within some domain.¹ The chain analysis used for concord among verbs in Hindi could in principle be applied to the nominal domain as well.

On the theoretical side, I adopt the mainstream Minimalist view that agreement involves heads with initially unvalued features, which obtain their value as part of the derivation (Chomsky 2000). This is fundamentally a practical choice, as it allows us to focus on the

¹Other distinctions include the fact that concord has a more local character, and that certain features are less likely to participate than others. However, there are almost always items to be skipped, so we cannot step down to a SL analysis of concord, and the latter issue is one of substance rather than form, and therefore of little interest to us here.

relations between the nodes themselves, which are where the relevant formal issues lie. The main alternative is a “checking” analysis, where all items are fully valued from the start, and agreement ensures that appropriate forms are inserted in positions where a choice can be made, such as that of finite T, as in early Minimalism (Chomsky 1995). Both of these are fundamentally compatible with a TSL-2 analysis, but the latter results in extremely verbose grammars, and brings to the forefront issues which are not directly relevant, such as whether third person is a distinct feature or the lack of a person feature (cf. Nevins 2007). It is interesting to note that this is essentially the opposite of the choice I made for case, where having the concrete case features from the start made for a more concise analysis.

Beyond this, I do not deal directly with most of the details of the theory of Agree, such as whether we also have a distinction between “interpretable” and “uninterpretable” features, and whether or not this aligns with the valued/unvalued distinction (cf. Pesetsky and Torrego 2007), or whether some or all of the process might actually be post-syntactic (cf. Bobaljik 2008; Kalin 2020). I also do not pay much attention to any particular version of the Agree algorithm, as represented by various iterations of cyclic Agree (Béjar and Rezac 2009) and the interaction-satisfaction theory (Deal 2015, 2024). Our focus is instead on how we can pair up agreeing elements correctly using a TSL-2 grammar. As we will soon see, we can get surprisingly far with no more than this.

6.3 The complement spine

We now add the couple of formal assumptions needed to get the TSL-2 analysis of agreement of the ground. As discussed in Chapter 2, c-strings were originally developed in order to handle licensing of individual nodes, such that the wellformedness of a dependency tree requires (among other things) that all c-strings are well-formed. This is appropriate for NPIs and bound pronouns/anaphora, which are licensed from above, but is slightly awkward for agreement where the dependent item (the probe) is usually the higher one. At the same time, we would like to enforce the c-command (or rather, d-command) constraint for agreement as well as relativized minimality, as both of these are highly characteristic of agreement. So, the tree tier model would not be much of an improvement.

What if we instead devised a type of downward-looking c-string, that is a string of *commanded* elements rather than a string of *commanding* elements? This would be closer to what we want, except that there is generally more than one such c-string, as the tree branches in the downward direction. This is highly problematic, since failure for one path to lead to the other half of the agreeing pair should not cause the derivation to be ruled out, but then the only way to determine if a probe is licensed is to check if *any* of its downward c-strings is licit. This is formally suspect since it adds a kind of quantification which is not present in the basic c-string model. Further, it makes the wrong empirical predictions, as syntactic dependencies tend not to occur between a head and an element deeply embedded in some adjunct or specifier (Graf and De Santo 2019). 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. This effect appears to

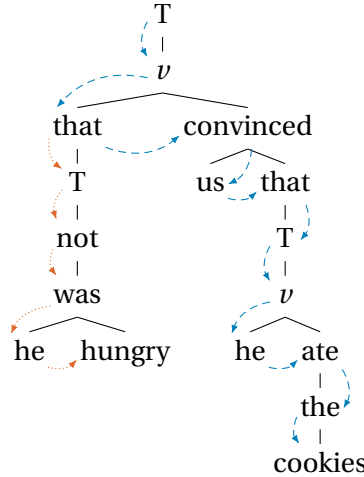


Figure 6.1: Tree and two spines for the sentence “[That he was not hungry] convinced us [that he ate the cookies]”. The subject clause begins a new spine (green dotted), while the complement clause shares a spine with the main clause (blue dashed)

be particularly robust for agreement, and I am unaware of any exceptions.²

Instead, agreement must follow the COMPLEMENT SPINE. While it can see the root of each specifier and adjunction, it can only recursively visit the contents of complements. Let us call a downward c-string which obeys the complement spine constraint a *spinal* c-string. To this, I add one further refinement. Rather than considering the spinal c-string of each node, we use only the string associated with the root of the tree. Collectively, the main spine and secondary spines will be simply called SPINES. Figure 6.1 illustrates the SPINES of the tree for [That he was not hungry] convinced us [that he ate the cookies], of which there are two. The MAIN SPINE is the path (T · *v* · that · convinced · us · that · T · *v* · he · ate · the · cookies), while the subject clause begins a SECONDARY 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*.

All of our TSL grammars for agreement will apply to spines. This solves the problem of multiple paths described above. In addition, it allows us to treat agreement in a way which closely parallels the treatment of movement in Chapter 2. As we will see in the next chapter, the parallel with long-distance phonotactic patterns becomes particularly clear in this system. And of course, we enforce the d-command constraint and relativized minimality.

There is a small but notable difference between this approach, on the one hand, and both tree tier and c-strings, on the other. While existing approaches to TSL syntax effectively deal with the two dimensions of the dependency tree by associating each node with a string, this one instead SEVERs the tree into a set of strings, each corresponding to a (nearly) disjoint

²Morphological agreement between a reflexive and its controller can occur even if the reflexive is embedded in an adjunct. This seems at first glance to violate our generalization, but such dependencies are upward looking, so the problem of multiple paths does not apply. See McFadden et al. (2019) for another take on this asymmetry which is similar in spirit to the present approach.

set of nodes. In this way, it is highly reminiscent of the model in Uriagereka (1999): the root of each complex specifier or adjunct is a privileged position where small amounts of information can cross; otherwise such branches are opaque. As discussed by De Santo and Graf (2019), an alternative would be to mark such nodes so that they can be distinguished in c-string grammars, and exceptions to the spine generalization allowed in special cases. I leave the exploration of such a model for future research.

6.4 Constraining agreement

Having established how to obtain the spines of the 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 above, 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. Following the terminology of Chomsky (2000), 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 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 $-\phi$.³ 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.

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 6.2. 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 $+\phi$ is immediately followed by one bearing $-\phi$, and that every element that bearing $-\phi$ is immediately preceded by one bearing $+\phi$. In 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 6.2 is repeated in (1a). (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 exam-

³Following 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 chapter.

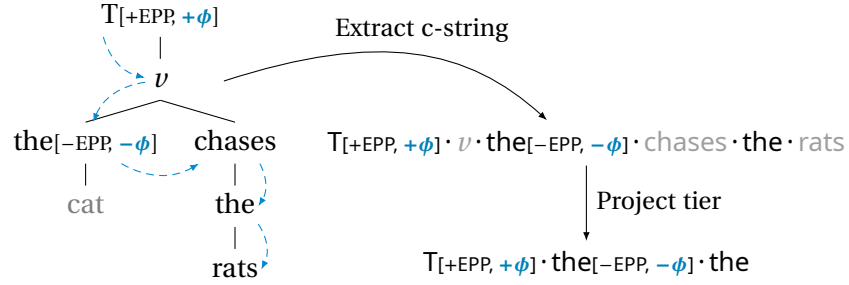


Figure 6.2: TSL analysis of English subject-verb agreement. Elements ignored at each step are grayed out

ple, it cannot bypass the subject in favor of agreement with the object, as in (1b). 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 $T_{[+\phi]} \cdot D$ and $D \cdot D_{[-\phi]}$ are illicit. Similarly, agreement with the object in (1c) is impossible because the subject C head intervenes, even though by hypothesis it cannot agree. Finally, agreement into CP is impossible (1d), though agreement into non-finite TP is possible; see (3) below.⁴

(1) Licit and illicit subject-verb agreement configurations

a. Subject agreement (✓)

- Sentence: The cat **chases** the rats.
- C-string: $T_{[+\phi]} \cdot \nu \cdot D_{[-\phi]} \cdot V \cdot D \cdot N$
- ϕ -agreement tier: $T_{[+\phi]} \cdot D_{[-\phi]} \cdot D$
- Constraints violated: n/a

b. Object agreement across DP (✗)

- Sentence: The cat **chase** the rats.
- C-string: $T_{[+\phi]} \cdot \nu \cdot D \cdot V \cdot D_{[-\phi]} \cdot N$
- ϕ -agreement tier: $T_{[+\phi]} \cdot D \cdot D_{[-\phi]}$
- Constraints violated: $T_{[+\phi]} \cdot D$, $D \cdot D_{[-\phi]}$

c. Object agreement across CP (✗)

- Sentence: [_{CP} That he plays the bassoon] **impress** us.
- C-string: $T_{[+\phi]} \cdot \nu \cdot C \cdot V \cdot D_{[-\phi]}$
- ϕ -agreement tier: $T_{[+\phi]} \cdot C \cdot D_{[-\phi]}$
- Constraints violated: $T_{[+\phi]} \cdot C$, $C \cdot D_{[-\phi]}$

d. Agreement into non-finite CP (✗)

- Sentence: It **are** possible [_{CP} for rats to have fleas.]
- C-string: $T_{[+\phi]} \cdot \text{be} \cdot A \cdot C \cdot \text{to} \cdot \nu \cdot D_{[-\phi]} \cdot V \cdot D \cdot N$

⁴I 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.

- 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 7.2.

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 $D_{[-\phi]} \cdot D_{[-\phi]}$.⁵ Putting all of this together, we arrive at the TSL-2 grammar shown in (2) 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 and Y are used as placeholders for elements 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.

(2) ϕ -agreement tier for English

- Project: all finite T , all D , all C
- Constraints: $\left\{ \begin{array}{ll} X_{[+\phi]} \cdot Y_{[+\phi]}, & X_{[-\phi]} \cdot Y_{[-\phi]}, \\ X_{[+\phi]} \cdot Y, & X \cdot Y_{[-\phi]}, \\ X_{[+\phi]} \cdot \times, & \times \cdot Y_{[-\phi]} \end{array} \right\}$

The overall analysis is extremely similar to the analysis of movement in Graf (2022c). 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 (3). For whatever reason, existential *there* seems to be invisible for agreement.

⁵It 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.

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.⁶ In addition, long-distance agreement across *there* is optional for many, if not most English speakers, an issue which we will revisit in Section 7.5.

(3) Long-distance agreement in existential sentences

- Sentence: There **seem** [_{TP} to be [_{PredP} some squirrels in the attic]].
- C-string: $T_{[+EPP, +\phi]} \cdot \nu \cdot \text{seem} \cdot \text{to} \cdot \text{be} \cdot \text{there}_{[-EPP]} \cdot \text{Pred} \cdot D_{[-\phi]} \cdot P \cdot D \cdot N$
- ϕ -agreement tier: $T_{[+\phi]} \cdot D_{[-\phi]} \cdot D$
- 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 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 2.3.4, 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 chapter.

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, just as we earlier assumed it to be independent from case. First and foremost, it would 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 2022b). 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

⁶I assume that *there* originates in Spec-*vP* 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 the Chapter 7, in which I survey a wide variety of agreement patterns from the syntactic literature and show that they are all TSL-2.

6.5 Summary

In this chapter, I introduced a TSL-2 model of agreement based on a restricted variant of a c-string, which I call a spine, in combination with MG-style diacritics representing probes and goals. The system was shown to encapsulate the basic locality properties which are desired of any theory of agreement. In the next chapter, I show that the natural parameters of the model correspond to attested cross-linguistic variation. From there, we will take a close look at the verbal agreement system of Hindi, where we can isolate two distinct processes involved in agreement, each TSL-2, which interact in such a way that produces an effect known as “parasitic agreement”.

Chapter 7

The Typology of Agreement

7.1 Overview

Graf (2022c) 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, using the model developed in the previous chapter. 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 7.2). In addition, TSL-2 also permits variation in directionality (Section 7.3), as well as seemingly complex configurations in which multiple probes share a single goal (Section 7.4) or a single probe interacts with multiple goals (Section 7.5). The chapter closes by revisiting the power of TSL-2 and its alignment with the observed typology (Section 7.6).

An overview of these parameters of the grammar and the corresponding agreement patterns is given in Table 7.1; the full set of patterns treated in this chapter is summarized in Table 7.2 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.

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 chapter. 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 2.3.5. However, I will sketch how this could be done in terms

This chapter is based on material from Hanson (2025c) “Tier-based strict locality and the typology of agreement”. There are no changes to the analysis. The only substantive addition is a set of examples of impossible constraints on agreement, mirroring the examples for movement and case presented in Chapters 2 and 4. I also updated the format of the TSL-2 grammars to use two variables X and Y as in the previous chapter.

Tier projection	Tier constraints	Phenomenon
a. All $\pm\phi$ elements	Strict matching of $+\phi/-\phi$	(Strict) Minimality
b. Some D heads do not project	(as in (a))	Invisibility
c. Some non-agreeing items 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

Table 7.1: Variants of a TSL-2 grammar and corresponding agreement phenomena

of the interaction-satisfaction theory (Deal 2015), applying the approach to the optionality problem mentioned above and discussing some of the caveats.

7.2 Invisibility and blocking

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 (1). I assume that nominative 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.

(1) 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

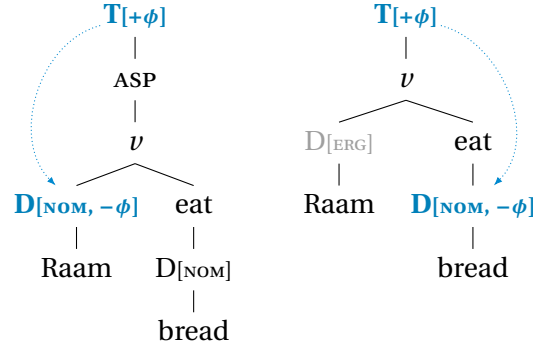


Figure 7.1: Case-sensitive agreement in Hindi. Ergative DPs are invisible, causing the object to agree instead of the subject

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 (1) are as shown in Figure 7.1. 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 7.4).

C-strings and their ϕ -agreement tiers for these examples are shown in (2) 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 (2a); if $-\phi$ is placed on the object (2b), the subject intervenes, resulting in a minimality violation. In the 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 (2c). Instead, it is the object that must agree, since it is adjacent to T on the tier (2d). This is also what happened in the example of agreement across existential *there* in Chapter 6.

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

a. Imperfective clause, subject agreement (✓)

- C-string: $T_{[+\phi]} \cdot \text{ASP} \cdot v \cdot D_{[\text{NOM}, -\phi]} \cdot V \cdot D_{[\text{NOM}]} \cdot N$
- ϕ -agr. tier: $T_{[+\phi]} \cdot D_{[\text{NOM}, -\phi]} \cdot D_{[\text{NOM}]}$
- Constraints violated: n/a

b. Imperfective clause, object agreement (✗)

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

- c. Perfective clause, subject agreement (✗)
- C-string: $T_{[+\phi]} \cdot \nu \cdot D_{[ERG, -\phi]} \cdot V \cdot D_{[NOM]} \cdot N$
 - ϕ -agr. tier: $T_{[+\phi]} \cdot D_{[NOM]}$
 - Constraints violated: $X_{[+\phi]} \cdot Y$
- d. Perfective clause, object agreement (✓)
- C-string: $T_{[+\phi]} \cdot \nu \cdot D_{[ERG]} \cdot V \cdot D_{[NOM, -\phi]} \cdot N$
 - ϕ -agr. tier: $T_{[+\phi]} \cdot D_{[NOM, -\phi]}$
 - Constraints violated: n/a

It should be noted that some Hindi verbs take dative subjects, and that these are also invisible.¹ In other words, it really is only nominatives that can agree. To summarize, the TSL grammar for Hindi is shown in (3). From this point forward, I will highlight what has changed in comparison to the English grammar from Chapter 6. In this case, only the tier projection rules have changed.²

(3) ϕ -agreement tier for Hindi

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

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 (4).

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

- a. Maile yas pasal-mā patrikāā 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.’

¹The 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 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.

²A reviewer of Hanson (2025c) 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.

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 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 (5).

(5) ϕ -agreement tier for Nepali

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

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 (2020b) for discussion.³

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, (6) is an example of agreement across a dative subject in a simple transitive clause, demonstrating invisibility; next, (7) 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 (7a) is not agreement with the dative DP, but a default ending.

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

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

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

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

³Note 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.

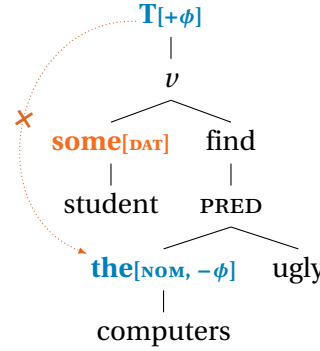


Figure 7.2: Dative intervention in Icelandic transitive expletive construction. Dative DPs do not agree, but also block agreement with a more distant DP

- b. * Pað finnst einhverjum stúdent [tölvurnar ljótar].
EXPL find.PL some student.SG.DAT 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 7.5 which could potentially be applied to the Icelandic data.

I assume the transitive expletive construction to involve a small clause structure, modelled here as a PredP, as shown in Figure 7.2. 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 lone probe without a goal, at least in this specific circumstance.⁴ 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 (8), with the relevant constraint modification highlighted. Next, c-strings and their tiers for the licit and illicit agreement configu-

⁴In 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.

rations in (7) are shown in (9). 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.

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

- Project: all T, all D, all C
- Constraints: $\left\{ \begin{array}{ll} X_{[+\phi]} \cdot Y_{[+\phi]}, & X_{[-\phi]} \cdot Y_{[-\phi]}, \\ X_{[+\phi]} \cdot \mathbf{D}_{[\text{NOM}]}, & X \cdot Y_{[-\phi]}, \\ X_{[+\phi]} \cdot \bowtie, & \bowtie \cdot Y_{[-\phi]} \end{array} \right\}$

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

a. Default agreement (✓)

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

b. Agreement across dative DP (✗)

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

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 chapter.

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 (6). This is shown in (10).

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

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

‘Some student finds the computers ugly.’

Based on this data, we would say that datives are invisible precisely when they undergo EPP movement.⁵ 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 (6), whose tier would contain $T_{[-\phi]}$ at the right edge of the tier, we also remove the constraint $X_{[+\phi]} \cdot \times$. The revised grammar is presented in (11) 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.

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

- Project: all T, **all D_[NOM]**, **D_[DAT] if not [–EPP]**, all C
- Constraints: $\left\{ \begin{array}{ll} X_{[+\phi]} \cdot Y_{[+\phi]}, & X_{[-\phi]} \cdot Y_{[-\phi]}, \\ X_{[+\phi]} \cdot \mathbf{D_{[NOM]}} & X \cdot Y_{[-\phi]}, \\ \mathbf{\cancel{X_{[-\phi]} \cdot \times}}, & \times \cdot Y_{[-\phi]} \end{array} \right\}$

The situation with visibility of dative DPs in Icelandic is essentially the opposite of the differential object marking pattern mentioned in Chapter 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.

7.3 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 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

⁵According to Kučerová (2016), the correct generalization is that dative DPs in Icelandic are invisible if they undergo object shift (movement to Spec- ν P), and intervene if they do not. This movement tends to be unavailable in the transitive expletive construction for semantic reasons.

concord is upward looking, as is case concord within the DP (if case is inherent on D, as we have been assuming).⁶

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 (12). In contrast, in Lubukusu (a Bantu language spoken in Kenya), we find upward agreement for noun class with the next higher subject (13). Note that not all complementizers agree in Lubukusu.

(12) 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.'

(13) 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.'
- b. Alfredi ka-bolela ba-ba-ndu [a-li ba-kha-khile].
c1.Alfred **c1**-said c2-c2-people **c1**-that 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 7.3 shows Diercks' analysis for (13a) 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

⁶Even if a harmony pattern is symmetric on the surface, as in the example in Section 2.3.4, 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.

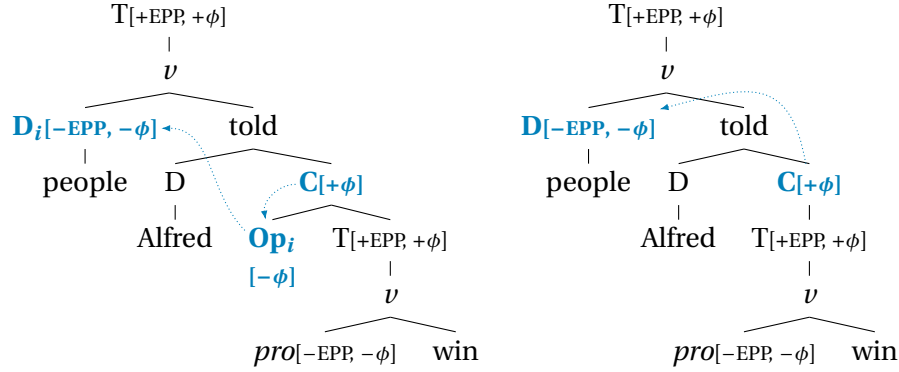


Figure 7.3: Complementizer agreement in Lubukusu. Left: indirect analysis. Right: direct agreement. In either analysis, one upward dependency is involved

is otherwise expected to agree. None of these arguments hold up. First, regarding subject-orientation, we have 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.⁷

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 on C is strictly subject-

⁷This 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 does feed subsequent operations.

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 2.3.5.

As usual, a selection of licit and illicit c-strings are given in (14), representing variants of example (13a). First, we have agreement with the upper subject, shown in (14a). 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 (14b) and the lower subject (14c), 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.

(14) C-strings and tier projections for Lubukusu

- a. Complementizer agrees with upper subject (✓)
 - C-string: $T_{[+EPP]} \cdot \nu \cdot \mathbf{D}_{[-EPP, -\phi]} \cdot V \cdot D \cdot \mathbf{C}_{[+\phi]} \cdot T_{[+EPP]} \cdot \nu \cdot D_{[-EPP]} \cdot V$
 - Tier: $\mathbf{D}_{[-EPP, -\phi]} \cdot \mathbf{C}_{[+\phi]} \cdot D_{[-EPP]}$
 - Constraints violated: n/a
- b. Complementizer agrees with upper object (✗)
 - C-string: $T_{[+EPP, +\phi]} \cdot \nu \cdot D_{[-EPP]} \cdot V \cdot \mathbf{D}_{[-\phi]} \cdot \mathbf{C}_{[+\phi]} \cdot T_{[+EPP, +\phi]} \cdot \nu \cdot D_{[-EPP]} \cdot V$
 - Tier: $D_{[-EPP]} \cdot \mathbf{C}_{[+\phi]} \cdot D_{[-EPP]}$
 - Constraints violated: $X \cdot C_{[+\phi]}$
- c. Complementizer agrees with lower subject (✗)
 - C-string: $T_{[+EPP]} \cdot \nu \cdot D_{[-EPP]} \cdot V \cdot D \cdot \mathbf{C}_{[+\phi]} \cdot T_{[+EPP]} \cdot \nu \cdot \mathbf{D}_{[-EPP, -\phi]} \cdot V$
 - Tier: $D_{[-EPP]} \cdot \mathbf{C}_{[+\phi]} \cdot \mathbf{D}_{[-EPP, -\phi]}$
 - Constraints violated: $X \cdot C_{[+\phi]}, X_{[-\phi]} \cdot \times$

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.⁸ Thus, our the grammar for Lubukusu is given in (15). Again, the primary change compared to the baseline English grammar is that the constraints have been mirrored.

(15) Complementizer ϕ -agreement tier for Lubukusu

- Project: $\mathbf{D}$ if $[-EPP]$, all C

⁸Failure 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.

- Constraints:

$$\left\{ \begin{array}{ll} X_{[+\phi]} \cdot Y_{[+\phi]}, & X_{[-\phi]} \cdot Y_{[-\phi]}, \\ \textcolor{blue}{X} \cdot \textcolor{blue}{Y}_{[+\phi]}, & \textcolor{blue}{X}_{[-\phi]} \cdot \textcolor{blue}{Y} \\ \textcolor{blue}{\times} \cdot \textcolor{blue}{Y}_{[+\phi]}, & \textcolor{blue}{X}_{[-\phi]} \cdot \textcolor{blue}{\times} \end{array} \right\}$$

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.

7.4 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 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.

Keeping with the theme of complementizer treatment, let us return to the example from West Flemish in (12b), repeated in (16).⁹ The structure assumed for this sentence is shown in Figure 7.4. 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 (17), 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.

- (16) Kvinden [_{CP} dan die boeken [_{DegP} te diere] zyn].
 I.find that.**PL** the book.**PL** too expensive be.**PL**

‘I find those books too expensive.’

⁹A very similar analysis can be used for concord phenomena of the sort mentioned in Section 7.3.

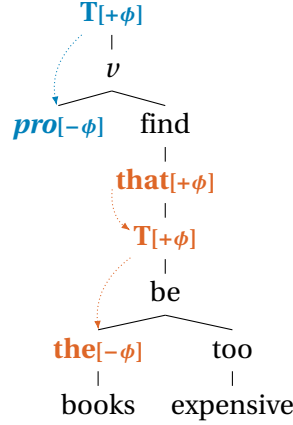


Figure 7.4: Embedded C and T share a goal in West Flemish. In such cases, the tier grammar permits a sequence of probes followed by a goal

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

- C-string: $T_{[+\phi]} \cdot v \cdot D_{[-\phi]} \cdot V \cdot C_{[+\phi]} \cdot T_{[+\phi]} \cdot v \cdot D_{[-\phi]} \cdot \text{DEG} \cdot A$
- ϕ tier: $T_{[+\phi]} \cdot D_{[-\phi]} \cdot C_{[+\phi]} \cdot T_{[+\phi]} \cdot D_{[-\phi]}$

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 $X_{[+\phi]} \cdot Y_{[+\phi]}$.¹⁰ Thus, the ϕ -agreement tier grammar for West Flemish is given in (18), with the removed constraint struck out.

(18) ϕ -agreement tier for West Flemish

- Project: all T, all D, all C
- Constraints: $\left\{ \begin{array}{ll} \cancel{X_{[+\phi]} \cdot Y_{[+\phi]}}, & X_{[-\phi]} \cdot Y_{[-\phi]}, \\ X_{[+\phi]} \cdot Y, & X \cdot Y_{[-\phi]}, \\ X_{[+\phi]} \cdot \bowtie, & \bowtie \cdot Y_{[-\phi]} \end{array} \right\}$

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 $X_{[+\phi]} \cdot X_{[+\phi, -\phi]}$ and $X_{[+\phi, -\phi]} \cdot X_{[-\phi]}$, but not $X_{[+\phi]} \cdot X_{[+\phi]}$. The trouble with this approach is that some elements may need to be lexically specified as being available either as $X_{[+\phi]}$ or $X_{[+\phi, -\phi]}$ 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.

¹⁰Note 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.

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 7.5. Before that, I present an example which 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 (19a) and object (19b) 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 (20).

(19) Dinka verb agreement with Spec-CP

- a. Mòc à-cé yĩn tĩŋ.
 man 3sg-PRF.SV you see.NF

‘The man has seen you.’

(Van Urk 2015, ch. 4, 19b)

- b. Yĩn Ø-cĩ môc tĩŋ.
 you 2-PRF.OV man.GEN see.NF

‘You, the man has seen.’

(Van Urk 2015, ch. 4, 20a)

(20) Agreement in both matrix and embedded clause

Yè kòc-kó [CP Op é-kè-yá ké tàak [CP è ____ é-kè-cĩ
 be **people**.CS-which PST-**PL**-HAB.2SG 3PL think.NF C PST-**PL**-PRF.OV
 Áyèn ké gâam gâlâm]]?
 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 +A' and + ϕ . 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 (2022c). A slightly

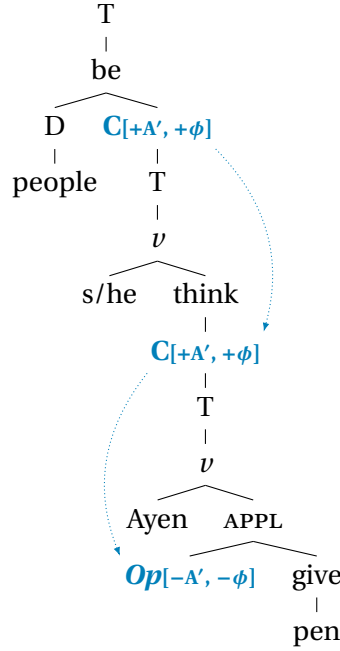


Figure 7.5: A'-agreement in Dinka. Each C head along the movement path agrees with the moving operator

simplified structure for example (20) which has been annotated accordingly is shown in Figure 7.5.¹¹

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 $-A'$. C also appears on the tier since it agrees, but T neither agrees nor blocks agreement, so it is omitted. The tier constraints are identical to our previous example of West Flemish. The full grammar is shown in (21), and example c-strings corresponding to sentences (19a) and (20) are shown in (22).

(21) ϕ -agreement tier for Dinka

- Project: **D** if $[-A']$, all C
- Constraints: $\left\{ \begin{array}{ll} \cancel{X_{[+\phi]}} \cdot \cancel{Y_{[+\phi]}}, & X_{[-\phi]} \cdot Y_{[-\phi]}, \\ X_{[+\phi]} \cdot Y, & X \cdot Y_{[-\phi]}, \\ X_{[+\phi]} \cdot \times, & \times \cdot Y_{[-\phi]} \end{array} \right\}$

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

a. Within-clause A'-movement

- C-string: $C_{[+A', +\phi]} \cdot T \cdot \nu \cdot D_{[-A', -\phi]} \cdot V \cdot D$
- Tier: $C_{[+A', +\phi]} \cdot D_{[-A', -\phi]}$

b. Long-distance A'-movement

¹¹The *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 νP ; in this case it is the latter.

- C-string: $T \cdot \text{be} \cdot D \cdot C_{[+A', +\phi]} \cdot T \cdot \nu \cdot D \cdot V \cdot C_{[+A', +\phi]} \cdot T$
 $\cdot \nu \cdot D \cdot \text{APPL} \cdot D_{[-A', -\phi]} \cdot V \cdot D$
- 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.

7.5 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 mentioned in Chapter 6, 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 (3), repeated below as (23a).¹² In contrast, agreement with the logical subject is obligatory if *there* is not inserted (23b).

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

- There **seem(s)** to be some squirrels in the attic.
- Some squirrels **seem(*s)** to be in the attic.

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 7.6. 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

¹²This 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*.

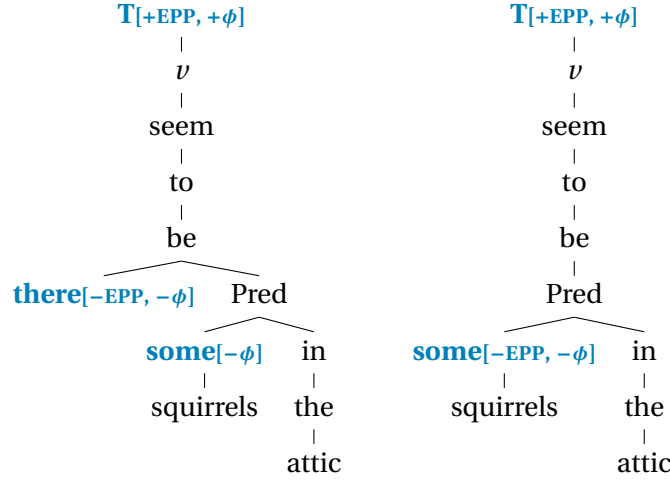


Figure 7.6: Interaction-satisfaction analysis of optional agreement in English existential clauses. Left: when *there* is inserted, T agrees both with *there* and the logical subject. Right: structure without *there*, where the logical subject is the only target for agreement

another ϕ -goal if the first goal is *there*, as shown in (24). As usual, c-strings for structures both with and without *there* are provided in (25). 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.

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

- Project: all T, all D, all C
- Constraints: as in (2), but allow $\text{there}_{[-\phi]} \cdot \text{D}_{[-\phi]}$

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

a. With *there*

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

b. Without *there*

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

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).¹³ Thus,

¹³Space 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 $\{P \cdot I, P \cdot S, P \cdot G, I \cdot I, I \cdot S, I \cdot G\}$ but not $\{S \cdot I, S \cdot G, G \cdot I, G \cdot G\}$. We must also distinguish *actual* interactors from *potential* interactors, perhaps with the same $-\phi$ diacritic used in this chapter.

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]. 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, $D[-\phi]$ may therefore be followed by another $D[-\phi]$ 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.).¹⁴

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 be 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.

7.6 Possible and impossible agreement patterns

To conclude this chapter, the agreement patterns analyzed above are summarized again in Table 7.2, 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 relativized minimality. For conciseness, only the Hindi variant of case-sensitive agreement is included; additionally, the sub-components of the Lubukusu and Dinka patterns 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 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

¹⁴The idea of treating extraction morphology as constraints on a movement path goes back at least to Bouma et al. (2001).

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 $-\phi$

Table 7.2: Summary of agreement patterns and their TSL-2 analysis

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 discussed in previous chapters regarding movement and case, 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.¹⁵ To drive this point home, consider the following set of hypothetical agreement systems:

- **Almost closest**
T can agree with the first or second closest DP, but not the third closest.
- **Mandatory intervention**
T can only agree with DP if exactly 1 horizon intervenes.
- **Rationed agreement**
At most n DPs in a sentence may agree.
- **Cowardly goals**
An ergative can agree only if there are at least n ergatives in the sentence.
- **Parity agreement**
When clauses are coordinated, agreement occurs in exactly one of them.
- **Introspective goals**
Only DPs which contain a relativized clause or a PP may agree.

As with the examples provided in Chapters 2 and 4, all of these are subregular, but they are not TSL-2, and hence ruled out for that reason. In stark contrast, the various obvious

¹⁵The 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.

manipulations of a TSL-2 grammar, such as mirroring the constraints, and allowing adjacent pairs of like elements, correspond to real movement, case, and/or agreement phenomena.

Even so, as several reviewers of Hanson (2025c) 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.¹⁶ It is for these reasons, among others, that I have stressed that the TSL-2 hypothesis is only one component of a complete theory. Finally, note that these issues are not at all specific to syntax; see Heinz (2010a) for a discussion of the situation in phonology.

7.7 Summary and discussion

In this chapter, 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.

Since this concludes our final typological investigation, now is a good time to review some of the recurring themes of this work, and some outstanding issues. First, it is worth reiterating that we do not expect every possible formal pattern to be attested due to the limited number of existing languages and the other factors influencing typology. As such, it is informative that so many predicted patterns represent actual agreement phenomena, just as we previously saw with movement and case. 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. The same can be said of the analysis of case using tree tiers. 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

¹⁶Regarding the role of language acquisition, 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, 2024) for an example from phonology and Hanson (2024d) regarding syntax.

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.

Next, 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. In Chapter 5, I also mentioned a potential issue that coordination can cause for tree tiers by breaking single case domain into several. Overall, this represents another major direction for future research.

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.

Chapter 8

Agreement in Hindi

8.1 Overview

Hindi-Urdu (HU) is famous for its verbal agreement system, which includes among its many interesting properties:

- case-sensitivity – only unmarked (nominative) DPs can agree
- long-distance agreement – agreement with an argument of an embedded verb
- default agreement – when there is no viable goal, default morphology results
- concord – infinitives and participles agree together with the finite verb

I discussed case-sensitivity in HU and other languages in the previous chapter, showing it to be just one manifestation of parametrized visibility as predicted by the TSL-2 model. We also witnessed long-distance agreement (LDA) and default agreement in the examples from Icelandic, though not in HU. Additionally, I noted the phenomenon of concord in the non-finite verbs, though I omitted an explicit treatment.

In this chapter, I provide a more comprehensive treatment of the HU data, including all of these phenomena as well as a new one: PARASITIC AGREEMENT. One might expect that concord always involves simple iterated agreement from one concordial item to the next, but this is not what happens in HU, at least not in all dialects, according to Bhatt (2005). Instead, any non-finite verbs (both participles and infinitives) agree with the target DP if and only if the finite verb of the same clause does, even if there is nothing in the structure to block agreement between the non-finite verbs and the target DP. In other words, our conception of agreement as a pairwise relation between syntactic nodes appears to break down.

Parasitic agreement is difficult to motivate under common theoretical assumptions. If infinitival verbs bear unvalued ϕ -features which probe for a value, then it should not matter whether or not some higher head also agrees with the same DP. If they do not bear such features, then why should they agree at all? Essentially, parasitic agreement makes no sense if agreement is driven solely by the need for the probe to obtain a value. Optional features are also of no help, because agreement on such items is not optional, but rather CONDITIONAL

This chapter is based on an analysis which was presented at SYNC 2025 (Hanson 2024a) and PLC 49 (Hanson 2025d).

on what happens elsewhere in the derivation. This is troubling from a formal perspective, as it sounds rather like an arbitrary boolean condition, that is, a locally testable pattern, which is beyond the capabilities of the class TSL and hence predicted not to exist.

The solution it turns out, is surprisingly simple. HU agreement involves not one but two tiers, each with a window of size two and each unremarkable in its own right. Thus, HU agreement is not TSL-2, but MTSL-2. Although this is probably not the first place one would look for the influence of multiple tiers in agreement, it is also not far fetched, as the two tiers correspond neatly to two separable processes which we might term AGREEMENT and CONCORD, respectively:

1. AGREEMENT: finite T agrees with the closest visible DP, if possible
2. CONCORD: all verbs along the path from T to DP agree iff T does

This analysis essentially implements Bhatt's (2005) intuition that all of the verbs are valued together when the probe on finite T finds a goal. Better yet, it explains why such a strange thing should ever happen. If language is capable of producing such patterns independently, we expect them to sometimes show up together; when they do, parasitic behavior of concordial items is the natural result.

The remainder of this chapter is structured as follows. I begin by reviewing the HU agreement data and descriptive generalizations, this time covering long-distance and parasitic agreement (Section 8.2). I then provide a comprehensive tier-based analysis as just described (Section 8.3). Finally, I revisit the typological question which was the focus of the previous chapter, and what the HU pattern adds to our understanding (Section 8.4).

8.2 Descriptive generalizations

Our first generalization is that the finite verb/auxiliary agrees with the highest nominative (unmarked) DP in its domain. This may be the matrix subject (1a), matrix object (1b), or embedded object (1c). (PRO, if present, is never a viable target for agreement.) Any infinitive/participle verbs agree with the same DP, but overtly case-marked arguments never agree. If there is no potential goal, default (MSG) agreement occurs on all verbs (1d). Note that all data in this section comes from Bhatt (2005). For clarity, verbal agreement morphology is highlighted in blue, default forms are in green, and agreeing DPs are underlined.

- (1) Agreement targets the highest accessible argument

- a. Subject agreement (unmarked subject/object)

Rahul kitaab paṛh-**taa** **thaa**
 Rahul.M book.F read-HAB.MSG be.PST.MSG

‘Rahul used to read (a/the) book.’

- b. Object agreement (ERG subject + unmarked object)

Rahul-ne kitaab paṛh-**ii** **thii**
 Rahul-ERG book.F read-PFV.F be.PST.FSG

‘Rahul had read the book.’

- c. LDA (ERG subject + unmarked embedded object)

Vivek-ne [kitaab parh-**nii**] chaah-**ii**
 Vivek-ERG book.F read-INF.F want-PFV.FSG

‘Vivek wanted to read the book.’

- d. Default agreement (ERG subject + ACC object)

Rahul-ne kitaab-ko parh-**aa** **thaa**
 Rahul-ERG book.F-ACC read-PFV.MSG be.PST.MSG

‘Rahul had read the book.’

This covers the properties of case-sensitivity, long-distance agreement, default agreement, and concord. Our second major generalization is that both participle agreement and infinitive agreement are PARASITIC on successful agreement by the finite verb: either all verbs agree or none do. This means that hypothetical versions of (1a–d) in which the finite verb and participle/infinitive act separately are impossible. Bhatt (2005) presents data only for the LDA case, shown below.

- (2) a. Infinitival agreement but no LDA

* Shahrukh-ne [tehnii kaat-**nii**] chaah-**aa** **thaa**
 Shahrukh-ERG branch.F cut-Inf.F want-PFV.MSG be.PST.MSG

- b. LDA but no infinitival agreement

* Shahrukh-ne [tehnii kaat-**naa**] chaah-**ii** **thii**
 Shahrukh-ERG branch.F cut-Inf.M want-PFV.F be.PST.FSG

‘Shahrukh wanted to cut the branch.’

Although these examples only demonstrate a mismatch between the embedded and matrix verbs, it is implied that the participle is also parasitic on the finite verb, and I am unaware of any counterexamples. From this point forward, I will not usually distinguish infinitives and participles since their behavior is the same, referring to them collectively as non-finite verbs.

We now deal with two small complications. First, LDA is blocked if the subject is visible. In this case, there is agreement only in the matrix clause; the infinitive agrees with neither the embedded object nor the matrix subject.

- (3) Local agreement blocks LDA

Shahrukh [tehnii kaat-**naa**] chaah-**taa** **thaa**
 Shahrukh branch.F cut-INF.M want-PFV.MSG be.PST.MSG

‘Shahrukh wants to cut the branch.’

This is not terribly surprising, but calls for a slight refinement of our definition of parasitic agreement: agreement occurs on all non-finite verbs along the *path* from the agreeing DP to the finite verb. Since the infinitive here is outside this path, it does not agree. The entire paradigm summarized in the table below.

The second complication is that LDA appears to be optional, at least at first glance.

Agreement target	Agreement on		
	Finite verb	Matrix participle	Embedded infinitive
Matrix sbj./obj.	yes	yes	no
Embedded object	yes	yes	yes
None	no	no	no

Table 8.1: Parasitic agreement in Hindi-Urdu.

- (4) a. LDA and infinitival agreement
 Ram-ne [roṭii khaa-**nii**] chaah-**ii**
 Ram-ERG bread.F eat-INF.F want-PFV.FSG
 ‘Ram wanted to eat bread.’
- b. No infinitival agreement, no LDA
 Ram-ne [roṭii khaa-**naa**] chaah-**aa**
 Ram-ERG bread.F eat-INF.M want-PFV.MSG
 ‘Ram wanted to eat bread.’

This optionally is only apparent. Bhatt (2005) shows that RESTRUCTURED clauses require LDA, while other infinitives block it; in addition, there is a subtle difference in interpretation, which I set aside. Following Keine (2019), I assume the former to be ν Ps and the latter to be TPs. In Keine’s terms, TP is a *horizon* for ϕ -agreement, but ν P is not (C is also a horizon in HU, as we assumed in Chapter 7). The data from (4) is reinterpreted below.

- (5) a. LDA across ν P
 Ram-ne [ν P roṭii khaa-**nii**] chaah-**ii**
 Ram-ERG bread.F eat-INF.F want-PFV.FSG
 ‘Ram wanted to eat bread.’
- b. No LDA across TP
 Ram-ne [τ P roṭii khaa-**naa**] chaah-**aa**
 Ram-ERG bread.F eat-INF.M want-PFV.MSG
 ‘Ram wanted to eat bread.’

We can now summarize the HU verbal agreement pattern as follows. Agreement occurs along a chain beginning with the finite verb and ending with a nominative DP. The chain can be broken by a horizon or another nominative DP, which case agreement must not occur. In slightly more formal terms:

(6) **Verbal agreement in HU**

Given a chain of d-commanding elements $V_{FIN} \cdot (V_{INF})^* \cdot DP_{NOM}$ which is not interrupted by a horizon or another nominative DP, V_{FIN} and all V_{INF} agree with DP_{NOM} . Verbs which are not part of such a chain do not agree.

Importantly, there is a dependency between the agreeing verbs and DP *as well as* between the finite verb and the non-finite verbs. In Bhatt's (2005) Minimalist analysis, finite T is the locus of agreement, and somehow mediates agreement between the DP goal and all verbs. In formal terms, we have constraints on two different tiers, since the dependency between T and D pays no attention to the verbs, while the other dependency includes them. I flesh out this idea in the next section.

8.3 The two-tier analysis

As mentioned in the introduction, HU agreement pattern can be decomposed into two interacting processes:

1. AGREEMENT: finite T agrees with the closest visible DP, if possible
2. CONCORD: all verbs along the path from T to DP agree iff T does

Since these two processes have different locality properties, they occupy two different tiers. This is schematized in Figure 8.1. On each tier, the relevant dependencies involve only adjacent elements, which means that a pair of TSL-2 grammars, or an MTSL-2 grammar, is sufficient. Let us go through the definition of each grammar, then show how together they derive the data discussed above.

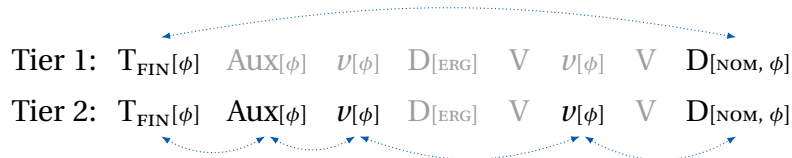


Figure 8.1: Two tiers for Hindi long-distance agreement. Only T and D appear on the first tier (“agreement tier”); on the second (“concord tier”), verbal elements appear as well.

8.3.1 The agreement tier

The agreement grammar is essentially the same as that proposed in Chapter 7. We project the agreeing elements (finite T and nominative D) as well as the blockers (non-finite T and C). The main change here is the addition of non-finite T.

- (7) **Agreement tier contents**
 $\{T_{\text{FIN}}, D_{\text{NOM}}, T_{\text{INF}}, C\}$

For illustration, I provide a worked example for the example of LDA in (1c) below. As before, I treat finite and non-finite verbs alike for purposes of agreement, taking transmission of tense to the highest verb to be a separate process. For simplicity, I have collapsed the probe and goal diacritics used earlier into a single agreement diacritic.

If we take the main complement spine and project the agreement tier, T and the highest nominative D are adjacent, allowing us to enforce agreement. In this way, subject agreement, object agreement, and now LDA are treated uniformly.

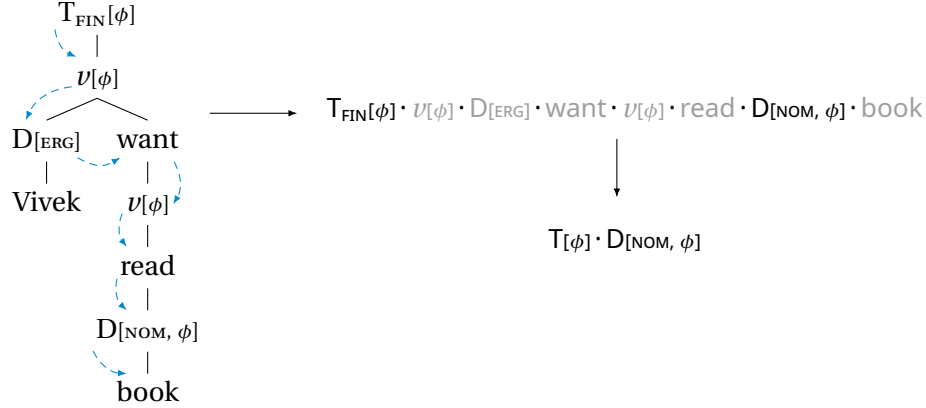


Figure 8.2: Tree, main spine, and agreement tier for (1c) *Vivek wanted to read the book*.

As for the constraints, the key bigrams can be summarized as follows:

(8) **Agreement tier constraints**

Probe must have a goal: $\{ *T_{FIN}[\phi] \cdot D[NOM], *T_{FIN}[\phi] \cdot C, *T_{FIN}[\phi] \cdot T_{INF}, *T_{FIN}[\phi] \cdot \times \}$

Goal must have a probe: $\{ *T_{FIN} \cdot D[NOM, \phi], *C \cdot D[NOM, \phi], *T_{INF} \cdot D[NOM, \phi], *D[NOM] \cdot D[NOM, \phi] \}$

Must agree when possible: $\{ *T_{FIN} \cdot D[NOM] \}$

In the present example, T immediately precedes the embedded object on the tier, so they agree, as required ($*T_{FIN} \cdot D[NOM]$). If a blocker intervenes ($*T_{FIN}[\phi] \cdot T_{INF}$), or there is no possible goal ($*T_{FIN}[\phi] \cdot \times$), agreement is ruled out. If there is a visible matrix subject, both object agreement and LDA are blocked, since the subject is higher ($*D[NOM] \cdot D[NOM, \phi]$). I provide complete derivations for all of the structures discussed above in Section 8.3.3.

8.3.2 The concord tier

In addition to T/D/C, the concord grammar also pays attention to the verbs. I will continue to assume that the agreeing part of the main verb (finite or non-finite) is the ν head. Additionally, the finite verb in some examples is an aspectual auxiliary, assumed to be of category Aux.¹

(9) **Concord tier contents**

$\{ T_{FIN}, D[NOM], \nu, \mathbf{Aux}, T_{INF}, C \}$

Next, recall our informal constraint, which is that the verbs agree iff they occur along a path from an agreeing T head to an agreeing D, that is, either all elements along the path agree or none of them do (skipping the irrelevant items). This is very easy to implement. We simply require that for each pair of elements on the tier along the path, either both bear the diacritic or neither does.

¹In the previous chapter this was labeled Asp. I change it to Aux here because the future auxiliary, which is not obviously an aspectual element, behaves the same way.

The key bigrams implementing this constraint are shown below. For brevity, constraints involving C are omitted; these would duplicate those with T_{INF} .

(10) **Concord tier bigrams implementing path constraint**

$$\left\{ \begin{array}{l} *T_{FIN[\phi]} \cdot Aux, \quad *T_{FIN[\phi]} \cdot \nu, \quad *Aux[\phi] \cdot \nu, \quad *\nu[\phi] \cdot \nu, \quad *\nu[\phi] \cdot D_{[NOM]}, \\ *T_{FIN} \cdot Aux[\phi], \quad *T_{FIN} \cdot \nu[\phi], \quad *Aux \cdot \nu[\phi], \quad *\nu \cdot \nu[\phi], \quad *\nu \cdot D_{[NOM, \phi]} \end{array} \right\}$$

There is one additional requirement, which is that elements in an incomplete chain must not agree. This happens when a blocker intervenes or there is no goal to be found.

(11) **Concord tier bigrams, cont.**

$$\{ * \nu[\phi] \cdot T_{INF}, *T_{INF} \cdot \nu[\phi], * \nu[\phi] \cdot \times \}$$

Notice all of the above bigrams have a simple and repetitive form: $*X_{[\phi]} \cdot Y$ or $*X \cdot Y_{[\phi]}$. To be clear, there are also some licit bigrams which violate this meta-constraint, specifically those that occur at the edge of a path. In particular, we can have $D_{[NOM, \phi]} \cdot D_{[NOM]}$, though not $*D_{[NOM]} \cdot D_{[NOM, \phi]}$, which is a minimality violation. Also note that some of these constraints are redundant in the sense that any derivation that violates them will also violate some constraint on the agreement tier. For this reason, I do not provide an exhaustive list.

8.3.3 Deriving the data

To briefly summarize the above analysis, we have two tiers, the first of which mandates that T agree with the closest DP if possible, while the second ensures that the verbs agree iff T does. Now, I demonstrate that the analysis correctly derives the target data while ruling out hypothetical versions which violate the descriptive generalizations. In the examples shown, the two TSL-2 grammars usually give the same result, making them semi-redundant, but not always. I also explain why a simple chain analysis on one tier does not work.

Licit configurations

We first show that the analysis rules in the licit agreement configurations, focusing on those in Table 8.2. For convenience, agreeing elements are highlighted in blue, concordial elements in green, and blockers in orange. The agreeing and blocking elements appear on both tiers, and the concordial appear on the concord tier.

To begin, (1a/1b/1c) represent licit cases of subject agreement, object agreement, and LDA, respectively. We take the LDA structure as representative.

(12) Derivation for LDA (1c)

Tier 1: $T_{FIN[\phi]} \cdot D_{[NOM, \phi]} \cdot D_{[NOM]}$	✓ Adjacent T and D agree
Tier 2: $T_{FIN[\phi]} \cdot \nu[\phi] \cdot \nu[\phi] \cdot D_{[NOM, \phi]} \cdot D_{[NOM]}$	✓ All verbs in chain agree

The next two cases involve incomplete chains. In (1d), all DPs are overtly case-marked, so non-agreement is allowed. In (5b), T_{INF} intervenes, creating two incomplete chains.

Ex.	Configuration	String (Key elements highlighted)
1a	Sbj. agreement	$T_{FIN}[\phi] \cdot \mathbf{Aux}[\phi] \cdot \nu[\phi] \cdot D_{[NOM, \phi]} \cdot \text{read} \cdot D_{[NOM]} \cdot \text{book}$
1b	Obj. agreement	$T_{FIN}[\phi] \cdot \mathbf{Aux}[\phi] \cdot \nu[\phi] \cdot D_{[ERG]} \cdot \text{read} \cdot D_{[NOM, \phi]} \cdot \text{book}$
1c	LDA	$T_{FIN}[\phi] \cdot \nu[\phi] \cdot D_{[ERG]} \cdot \text{want} \cdot \nu[\phi] \cdot \text{read} \cdot D_{[NOM, \phi]} \cdot \text{book}$
1d	Dflt. (No goal)	$T_{FIN} \cdot \mathbf{Aux} \cdot \nu \cdot D_{[ERG]} \cdot \text{want} \cdot \nu \cdot \text{read} \cdot D_{[ACC]} \cdot \text{book}$
5b	Dflt. (Blocked)	$T_{FIN} \cdot \nu \cdot D_{[ERG]} \cdot \text{want} \cdot T_{INF} \cdot \nu \cdot \text{read} \cdot D_{[NOM]} \cdot \text{book}$
3	Blocked by sbj.	$T_{FIN}[\phi] \cdot \mathbf{Aux}[\phi] \cdot \nu[\phi] \cdot D_{[NOM, \phi]} \cdot \text{want} \cdot \nu \cdot \text{cut} \cdot D_{[NOM]} \cdot \text{branch}$

Table 8.2: Licit agreement and non-agreement configurations.

(13) No potential goal

Tier 1: T_{FIN} ✓ No D to agree with

Tier 2: $T_{FIN} \cdot \mathbf{Aux} \cdot \nu \cdot \nu$ ✓ All non-agreeing

(14) Clause boundary intervenes

Tier 1: $T_{FIN} \cdot T_{INF} \cdot D_{[NOM]}$ ✓ Non-adjacent T and D don't agree

Tier 2: $T_{FIN} \cdot \nu \cdot T_{INF} \cdot \nu \cdot D_{[NOM]}$ ✓ Each chain is consistently non-agreeing

In the final configuration, the subject is nominative, so LDA with the embedded object is blocked. This creates two chains, the first complete, and the second incomplete. We therefore correctly predict that only the verbs in the first chain agree.

(15) Subject blocks LDA

Tier 1: $T_{FIN}[\phi] \cdot D_{[NOM, \phi]} \cdot D_{[NOM]}$ ✓ Lower D does not agree

Tier 2: $T_{FIN}[\phi] \cdot \mathbf{Aux}[\phi] \cdot \nu[\phi] \cdot D_{[NOM, \phi]} \cdot \nu \cdot D_{[NOM]}$ ✓ Lower chain non-agreeing

Illicit agreement configurations

Now, we consider key structures that should be ruled out, shown in Table 8.3. We ignore chains with pairs containing mismatched diacritics, whose ungrammaticality is straightforwardly ruled out by Tier 2.

In (1d'), T/Aux/ ν agree even though there is no visible DP. This violates constraints on both tiers.

(16) Agreement with nothing

Tier 1: $T_{FIN}[\phi]$ ✗ Probe has no goal

Tier 2: $T_{FIN}[\phi] \cdot \mathbf{Aux}[\phi] \cdot \nu[\phi] \cdot \nu[\phi]$ ✗ No D at end of chain

5b' is a locality violation. T_{INF} intervenes, breaking the chain. This violates constraints on both tiers.

Ex.	Configuration	String (Tier elements highlighted)
1a'	No subj. agreement	$T_{FIN} \cdot \text{Aux} \cdot v \cdot D_{[NOM]} \cdot \text{read} \cdot D_{[NOM]} \cdot \text{book}$
1b'	No obj. agreement	$T_{FIN} \cdot \text{Aux} \cdot v \cdot D_{[ERG]} \cdot \text{read} \cdot D_{[NOM]} \cdot \text{book}$
1c'	No LDA	$T_{FIN} \cdot v \cdot D_{[ERG]} \cdot \text{want} \cdot v \cdot \text{read} \cdot D_{[NOM]} \cdot \text{book}$
1d'	Agreeing with nothing	$T_{FIN}[\phi] \cdot \text{Aux}[\phi] \cdot v[\phi] \cdot D_{[ERG]} \cdot \text{want} \cdot v[\phi] \cdot \text{read} \cdot D_{[ACC]} \cdot \text{book}$
5b'	Locality violation	$T_{FIN}[\phi] \cdot v[\phi] \cdot D_{[ERG]} \cdot \text{want} \cdot T_{INF} \cdot v[\phi] \cdot \text{read} \cdot D_{[NOM, \phi]} \cdot \text{book}$

Table 8.3: *Illicit* agreement and non-agreement configurations, inverting the whether agreement occurs compared to the licit configurations.

(17) Locality violation

- Tier 1: $T_{FIN}[\phi] \cdot T_{INF} \cdot D_{[NOM, \phi]}$ ✗ Probe and goal not adjacent
Tier 2: $T_{FIN}[\phi] \cdot v[\phi] \cdot T_{INF} \cdot v[\phi] \cdot D_{[NOM, \phi]}$ ✗ Agreement in incomplete chains

Finally, in (1a'/1b'/1c'), agreement fails to occur where it should. This is ruled out by Tier 1. Importantly, there are *no violations* by Tier 2. Once again, we take (1c') as representative.

(18) Illicit lack of LDA

- Tier 1: $T_{FIN} \cdot D_{[NOM]} \cdot D_{[NOM]}$ ✗ T and D are adjacent but don't agree
Tier 2: $T_{FIN} \cdot \text{Aux} \cdot v \cdot D_{[NOM]} \cdot D_{[NOM]}$ ✓ All non-agreeing

8.3.4 Why a single-tier analysis does not work

It is obvious why Tier 1 alone is insufficient to handle the HU data: the verbal elements are omitted from the tier, so their distribution cannot be restricted at all. But Tier 2 on its own is also insufficient. It can rule in the licit structures, and rule out agreement where it should not occur, but it cannot *mandate* agreement when it should occur.

The immediately proceeding example shows why. We must allow pairs of non-agreeing elements at all positions in the chain since they occur in licit structures where agreement fails. The licit counterparts to the bigrams in (10) are listed below.

(19) *Licit* bigrams on Tier 2

$$\left\{ \begin{array}{ccccc} T[\phi] \cdot v[\phi], & T[\phi] \cdot \text{Aux}[\phi], & \text{Aux}[\phi] \cdot v[\phi], & v[\phi] \cdot v[\phi], & v[\phi] \cdot D_{[NOM, \phi]}, \\ T \cdot v, & T \cdot \text{Aux}, & \text{Aux} \cdot v, & v \cdot v, & v \cdot D_{[NOM]} \end{array} \right\}$$

But once we allow this, complete chains of non-agreeing elements are always allowed. For example, all of the pairs in (1c') — $T_{FIN} \cdot v \cdot v \cdot D_{[NOM]}$ — occur in (1d) or (5b). Non-agreement in such a structure is therefore incorrectly predicted to be possible.

In general, non-local dependencies cannot be reduced to chains of local links. This principle goes far beyond the present example—it strikes at the heart of why tiers are necessary to begin with. As soon as the constraint window moves past the initial item in the chain, it completely forgets what it has seen. However, such an iterative process can exist alongside

another which directly connects the two ends of the chain. This is what the HU agreement pattern is from a formal perspective.

This, in turn supports Bhatt’s key insight, which is that T mediates agreement between the verbs and the target DP. Only T appears on both tiers, allowing information to be transmitted in a non-linear manner. At the same time, no overt coordination is necessary. Each tier simply rules out derivations that violate its constraints. What we call parasitic agreement is just the natural result of applying both sets of constraints simultaneously.

More broadly, these results support the idea that Hindi LDA involves two independent processes, assuming as we have been that individual syntactic processes are TSL-2; these in turn combine via simple intersection to produce an MTSL-2 grammar. Importantly, we did not need to construct convoluted tier alphabets or constraints in order to ensure the correct result, instead, the tier grammars that encode the relevant descriptive generalizations just look like minor variants of patterns witnessed in the previous chapter. I expand on this point in the following section.

8.4 The typology of agreement, revisited

In Chapter 7, I focused on how the natural parameters of the TSL-2 model of agreement correspond closely to attested typological variation. Several of these case studies involved visibility conditioned by another feature, such as case, which is itself regulated by a different tier. The data in this chapter is slightly different, as it involves an interaction between two processes involving verbal agreement, one more local than the other, which results in what has been termed parasitic agreement. This study therefore represents a first step into the realm of multi-tier agreement processes, which as I mentioned in previous chapters are common and varied enough to warrant their own dedicated study. That said, in line with the theme of this work, we expect to see variants of the present pattern in different languages, as well as similar patterns in other linguistic domains. Let us briefly consider some representative examples.

8.4.1 Related languages and dialects of Hindi-Urdu

As discussed by Bhatt (2005), there are related languages (as well as dialects of HU) which are similar with respect to verbal agreement except that infinitive agreement is not parasitic on matrix clause agreement. This means that the embedded infinitive can agree with the object while the matrix verbs agree with the subject in structures like (3).

We could analyze such languages as follows. Infinitives and participles can always agree with closest visible DP. Higher heads agree with lower heads, up to T_{FIN} (which agrees) or T_{INF} (which does not). Formally, just a single tier, very similar to our concord tier, is required. The constraints are mostly unchanged, with two key differences. First, $T_{\text{INF}} \cdot \nu_{[\phi]}$ is allowed, since agreement with a higher finite verb is not required. Second, $*\nu \cdot D_{[\text{NOM}]}$ is banned, since agreement is (according to my understanding) still obligatory. The latter constraint effectively takes over the primary role of our agreement tier, which is to ensure that agreement occurs when a potential goal is available.

8.4.2 English affix hopping

We actually need two tiers even for the traditional generative analysis of English: finite T, not the verb, agrees with the subject, and a second process of AFFIX HOPPING transmits both tense and agreement morphology onto the highest verb, if possible. The present analysis of HU essentially implements the version in Adger (2003), which treats the second step as agreement. The key differences are that in HU, i) only nominatives are visible, and ii) agreement in the second step is chained. But these are just two of the basic parameters of the TSL-2 model of agreement! If this analysis for English is on the right track, then parasitic agreement in HU is even less surprising, since two tiers are required in either case. For English, T can see across a Neg head to agree with the subject, but cannot transmit its features to the verb across the same head.

That said, affix hopping could also be viewed as post-syntactic. This might be more faithful to Chomsky's (1957) original analysis, depending on how one interprets the transformation used to manipulate the order of morphemes. Perhaps the concord process in HU could also be viewed in this way, even if agreement with finite T is not. By analogy with the suggestion I made in Chapter 3 with respect to case, this corresponds to a system in which agreement on the verbal nodes is handled entirely within the mapping from the dependency tree to the corresponding phrase structure tree, using only the agreement features on T and D as a guide. I leave an exploration of this idea to future work.

8.4.3 Extraction morphology

Extraction morphology as analyzed by Graf (2022a) is similar to parasitic agreement from the present perspective. Once again, two tiers are generally needed. The first pairs movers and landing sites, while the second places constraints on the form of certain elements along this path. Here too, the two tiers may be collapsed in certain special cases, but usually the problem related to pure chain analyses noted above prevents this from working. One notable difference between these phenomena is that while agreement is allowed to fail (Preminger 2014), movement usually cannot.

8.4.4 Phonology

Looking beyond syntax, some unbounded circumambient processes (Jardine 2016; McCollum et al. 2020) might be similar to parasitic agreement. Readers may have noticed that there is a certain indeterminism in the directionality of parasitic agreement, as the verbs depend on information coming from the DP below as well as the T head above. Similarly, some phonological processes depend on information coming from both directions.

One complication in making the comparison precise is that existing work on such patterns in subregular phonology has focused on the map from UR to SR rather than constraints on structures or derivations. This matters for several reasons, but to mention just one, a process may be clearly directional in the map even if it is symmetric on the surface.

In any case, we can also see similar points of variation among harmony processes in phonology, since in addition to bidirectional processes we also have local and non-local

unidirectional processes. Furthermore, we can categorize the elements in both domains in similar ways. In HU, *v* and Aux like normal harmonizing segments, T_{INF} is a blocker, and T_{FIN} is like an “icy target” which agrees but terminates harmony.

8.5 Summary

In this chapter, I provided an analysis of verbal agreement in Hindi-Urdu which involves two simple TSL-2 grammars, corresponding to two logical processes with different locality profiles. Parasitic agreement is shown to be an epiphenomenon, and we now have a clear formal basis for the idea that T mediates agreement between a DP and the agreeing verbs, not the other way around (cf. Bhatt 2005). This study also provides an excellent example of why it is not generally possible to reduce long-distance dependencies to local links, which helps explain we need tiers in the first place. More broadly, parasitic agreement in HU is just a variant of phenomena found in other languages and in domains outside of agreement, just as predicted by the TSL-2 model.

Chapter 9

Conclusion

In the preceding chapters, I showed how the TSL-2 model of syntactic dependencies accounts for a broad range of data in the realms of case and agreement in a simple and unified manner. Having seen how the model works in detail, let us review what we have learned.

To begin, we surveyed a few representative classes of subregular languages, and how they provide a good upper bound for most phonological patterns. The SL formal languages correspond to local dependencies, and the superclass TSL subsumes most non-local dependencies. Of particular interest is the class TSL-2, as non-local dependencies seem not to need the ability to see more than two elements at a time. This provides a precisely defined model of linguistic locality which corresponds closely with observed generalizations. We then shifted our attention to syntax. Using the MG dependency tree formalism as a starting point, I showed how long-distance processes such as movement are also naturally understood as involving TSL-2 computations. Similar to phonology, this is not just an upper bound, but actually provides insight into the precise range of phenomena which are attested. In this way, subregular syntax succeeds where previous approaches based on surface strings fell short.

There were a couple of formal complications which we dealt with during the investigation. First, there are multiple reasonable ways of generalizing the TSL-2 languages to trees. Of these, two major approaches have been explored: tree tiers and tiers over command strings. From everything we know at present, it is unclear which, if either of these, is the “right” approach. Rather than attempting to answer this question directly, I decided to adopt each in the context where it seems most natural, selecting tree tiers for case and command strings for agreement. In the future, it would be useful to determine how effective each approach is in the opposite domain. The second complication is that the grammar as a whole is not TSL, but rather MTSL, since different dependencies clearly have different locality profiles and visibility conditions. This is particularly true of syntax, and exposes a possible weakness to the restrictiveness claim, as MTSL intuitively feels much less constrained. Nonetheless, there is reason to believe that restrictiveness can be maintained once we embed the formal model in a larger theory, including other formal constraints, linguistic substance, and the learning process, among others. In any case, the first step is to demonstrate the adequacy of the TSL model for individual dependencies—analysis before synthesis.

This brings us to the main contributions of this dissertation. I first showed how case dependencies can be understood as a relation between domain nodes and DPs on a tree tier,

with a window of size two in both dimensions. In this respect, case is extremely similar to movement, except that the range of daughter string languages is more diverse, as case makes unusually heavy use of the information in the relation between adjacent sisters on the tier. This is the essence of both dependent case assignment and differential argument marking. We also examined examples of locality effects and iteration, which closely mirror those found in movement. Might some of the more exotic aspects of case have parallels elsewhere? There is evidence of this in certain very complex agreement systems (cf. Clem and Deal 2024), but perhaps movement deserves a second look as well.

Agreement, in contrast to case, was modeled as in terms of adjacent pairs of elements along a tier projected from the complement spine—a restricted version of the command string model. Though slightly different in setup from tree tiers, only TSL-2 computations are needed in each case. Since we are dealing primarily with a pair of diacritics in a single dimension, the parameters of the model, and their corresponding empirical predictions, are especially clear. Visibility may be conditioned by a variety of factors, direction may be inverted, and chaining may be allowed or barred. We also saw many examples of interactions with case and agreement, especially with respect to visibility. It may well be no coincidence that so many of these interactions can be handled without the added complications found in many theories.

At several points, we reflected on the ways in which linguistic locality shows the same basic form across disparate phenomena representing multiple modules of grammar. Blockers of long distance harmony, syntactic islands, horizons, case sensitivity, and more were revealed to be manifestations of the same underlying mechanism of tier projection. Beyond this, similar variation can be found among the constraints: progressive and regressive harmony, different case alignments, and downward/upward agreement are related to the relative ordering of elements within the constraint window; iterated harmony, multiple *wh*-movement, case spreading, and chain agreement are the result when the constraints allow successive instances of the same element label; and so on. While I did not conduct a systematic study of the degree to which each type of pattern occurs in different domains, it is clear that the basic predictions of the TSL-2 model are well attested.

I also discussed some of patterns which we do *not* see, and how they are ruled out by the TSL-2 model. I clarified that while not every unattested pattern can be ruled out in this manner, many logically simple patterns provably can be, especially when we take care to avoid add unmotivated abstraction into the structural analysis, which is a practice that linguists often (though not always) strive to follow in any case. As a result, the model achieves a high degree of combined empirical coverage and restrictiveness with very few formal assumptions. Overall, this work adds further support to the claims made by McMullin and Hansson (2016) and Graf (2022c) in the context of long-distance phonotactics and movement.

While most of the case studies were very brief, we also looked in greater depth into a couple of languages, focusing on case assignment in Japanese and long-distance agreement in Hindi. This served several important purposes. First, it demonstrates the viability of the model in accounting for the overall structure of individual languages, not just small fragments in isolation. In doing so, we were able to make sense of a remarkable range of properties of each language, exposing the underlying simplicity of the system. Second, it allowed us to

explore some characteristics of a multi-tier model. The present studies were limited to three tiers for case in Japanese, and two for agreement in Hindi. This is a far cry from the minimal number of tiers which a complete syntactic analysis demands, but this is the first time that subregular analyses of even this scale have been attempted. As such, this dissertation lays the groundwork for future studies of the suitability of an MTSL model of syntax.

Together with past results on movement, we now have substantial evidence that long-distance syntactic dependencies are in general TSL-2, with binding as a possible outlier. Combined with work on long-distance phonotactics, this gives us good reason to believe that TSL-2 computations encapsulate a fundamental aspect of human language which plausibly derives from the specific nature of the underlying cognitive system.

Although I have hardly discussed morphology and semantics, there is some evidence that they too are based on TSL computations. For example, Aksënova et al. (2016) found that morphotactic restrictions are at most TSL, and Graf (2024b) argues that the meanings of lexical quantifiers can be understood in TSL-2 terms. There is also a large body of work on phonological maps, that is, the relation between underlying and surface forms, which finds that they are largely SL (Chandlee 2014; Chandlee and Heinz 2018) or TSL (Burness et al. 2021). Additionally, Chandlee (2017) shows that a wide range of morphological maps are at most SL, and Graf (2023) shows that a TSL tree-to-tree map is enough to implement movement. It is not yet known whether the subclass TSL-2 stands out as clearly in characterizing long-distance processes in these other domains, but this is now a prime subject for investigation.

So then, what comes next for subregular syntax? First of all, the empirical investigation which I have advanced in this dissertation is in no way complete. There may well be syntactic phenomena which defy the TSL-2 model (aside from binding), just as a small number of exceptions have been identified in phonology (De Santo and Graf 2019; Mayer and Major 2018; Graf and Mayer 2018). Furthermore, as I have emphasized, the purpose of subregular linguistics is not merely to establish complexity results, but to derive empirical and theoretic insights through the formal lens. There are many outstanding puzzles in syntax alone, and the present approach could provide the fresh perspective which is needed to solve them.

Second, we must closely examine the typological adequacy of the MTSL-2 model. This involves both an exploration of multi-tier patterns within a single domain, such as those covered in the chapters on Japanese and Hindi. In particular, agreement patterns which involve multiple exponence, (partial) dissociation of person/number/gender, person hierarchy effects, and so on, will serve as an ideal testing ground for the MTSL hypothesis: can all constraints be handled with a small number of independent, well-motivated tiers? A more systematic investigation of interactions between movement, case, and agreement is also called for. There is a strong possibility that many of complications of existing theories can be eliminated if we can determine the basic class of patterns that must be accounted for, since we would be able to design them around an appropriate universal formal generalization.

Third, there are several formal issues which are entangled with the TSL-2 hypothesis but logically separate, and which deserve further exploration. As mentioned at various points in this dissertation, we should determine whether case can be insightfully modeled using command strings, and agreement with tree tiers. Similarly, we should explore whether some or all aspects of these phenomena can be modeled using subregular tree-to-tree maps, perhaps

using the system of Graf (2023). Alternatively, Ikawa and Jardine (2021) combine commands strings with subregular string-to-string maps in their model of binding. There are no doubt other productive ways to think about the subregular complexity of syntax which have not yet been considered. For example, can the derivations of other syntactic formalisms such as TAG (tree adjoining grammar) and CCG (combinatory categorial grammar) also be modeled using only TSL?

Last but not least, we should begin to seriously consider the role of learning in subregular syntax. There are already many results demonstrating the efficient learnability of TSL string languages (Jardine and Heinz 2016; Jardine and McMullin 2017; Lambert 2021) as well as subclasses of MTSL (McMullin et al. 2019) and SS-TSL (De Santo and Aksënova 2021; Johnson and De Santo 2024). In principle, these can be adapted to tree languages (Swanson 2024), but such work is in its infancy. However, the practical learnability of TSL-2 patterns over command strings (or ancestor strings) can already be tested—all that is needed is an appropriate data source from which these strings can be extracted. Related to this, we should also leverage existing work on the construction of linguistic generalizations, such as that using the Tolerance Principle (Yang 2016); also see Belth (2023, 2024), Heuser et al. (2024), and Hanson (2024d). These two aspects of learning each provide something that other lacks; as such, they are a natural fit. There are many other interesting and worthwhile directions that this research program could go, but these seem to me to be the most obvious next steps.

In summary, I have established the subregular complexity of case and agreement: individual dependencies are TSL-2 over MG dependency trees, and the combined system, also including movement, is plausibly MTSL-2. This fills a major gap in our understanding of the subregular complexity of syntax, setting the stage for a shift from preliminary case studies towards increasing depth and integration.

Bibliography

- Abney, Steven P. (1987). “The English noun phrase in its sentential aspect”. PhD thesis. Cambridge, MA: Massachusetts Institute of Technology.
- Adger, David (2003). *Core syntax: A Minimalist approach*. Vol. 20. Core Linguistics. Oxford University Press.
- Adger, David (2025). *Mereological syntax: Phrase structure, cyclicity, and islands*. Cambridge, MA: MIT Press.
- Aksënova, Alëna and Sanket Deshmukh (2018). Formal Restrictions On Multiple Tiers. *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*. Ed. by Gaja Jarosz, Brendan O’Connor, and Joe Pater. Amherst, MA: University of Massachusetts Amherst, pp. 64–73. DOI: 10.7275/R5K64G8S.
- Aksënova, Alëna, Thomas Graf, and Sedigheh Moradi (2016). Morphotactics as Tier-Based Strictly Local Dependencies. *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pp. 121–130. URL: <https://www.aclweb.org/anthology/W/W16/W16-2019.pdf>.
- Andrews, Avery (1982). The representation of case in Modern Icelandic. *The mental representation of grammatical relations*. Ed. by Joan Bresnan. Cambridge, MA: MIT Press, pp. 427–503.
- Aoun, Joseph and Dominique Sportiche (1982). On the formal theory of government. *The Linguistic Review* 2.3, pp. 211–236. DOI: 10.1515/tlir-1983-020303.
- Backofen, Rolf, James Rogers, and K. Vijay-Shanker (1995). A first-order axiomatization of the theory of finite trees. *Journal of Logic, Language and Information* 4.1, pp. 5–39. DOI: 10.1007/bf01048403.
- Baker, Mark C. (2008). *The syntax of agreement and concord*. Cambridge Studies in Linguistics 115. Cambridge: Cambridge University Press. DOI: 10.1017/CB09780511619830.
- Baker, Mark C. (2015). *Case*. Cambridge Studies in Linguistics 146. Cambridge University Press. DOI: 10.1145/1463891.1464000.
- Baker, Mark C. and Nadya Vinokurova (2010). Two modalities of case assignment: Case

- in Sakha. *Natural Language & Linguistic Theory* 28.3, pp. 593–642. DOI: 10.1007/s11049-010-9105-1.
- Bárány, András (2017). *Person, Case, and Agreement*. Oxford University Press, Nov. DOI: 10.1093/oso/9780198804185.001.0001.
- Bárány, András and Michelle Sheehan (2024). Challenges for dependent case. *The Place of Case in Grammar*. Oxford, UK: Oxford University Press, July, pp. 93–126. DOI: 10.1093/oso/9780198865926.003.0005.
- Béjar, Susana and Arsalan Kahnemuyipour (2017). Non-canonical agreement in copular clauses. *Journal of Linguistics* 53.3s, pp. 463–499. DOI: 10.1017/s002222671700010x.
- Béjar, Susana and Milan Rezac (2009). Cyclic agree. *Linguistic Inquiry* 40.1, pp. 35–73. DOI: 10.1162/ling.2009.40.1.35.
- Belth, Caleb (2023). Towards a Learning-Based Account of Underlying Forms: A Case Study in Turkish. *Proceedings of the Society for Computation in Linguistics (SCiL) 2023*. Amherst, MA: University of Massachusetts Amherst, pp. 332–342. DOI: 10.7275/BC8Q-VJ22.
- Belth, Caleb (2024). A Learning-Based Account of Phonological Tiers. *Linguistic Inquiry*, pp. 1–37. DOI: 10.1162/ling_a_00530.
- Berro, Ane and Ricardo Etxepare (2017). Ergativity in Basque. *The Oxford Handbook of Ergativity*. Ed. by Jessica Coon, Diane Massam, and Lisa Demena Travis. Oxford University Press, Aug., pp. 782–806. DOI: 10.1093/oxfordhb/9780198739371.013.32.
- Bhatt, Rajesh (2005). Long Distance Agreement in Hindi-Urdu. *Natural Language & Linguistic Theory* 23, pp. 757–807. DOI: 10.1007/s11049-004-4136-0.
- Bickel, Balthasar and Johanna Nichols (2008). Case Marking and Alignment. *The Oxford Handbook of Case*. Ed. by Andrej L. Malchukov and Andrew Spencer. Oxford University Press, pp. 304–321. DOI: 10.1093/oxfordhb/9780199206476.013.0021.
- Bittner, Maria and Ken Hale (1996). The Structural Determination of Case and Agreement. *Linguistic Inquiry* 27.1, pp. 1–68. URL: <http://www.jstor.org/stable/4178925>.
- Blake, Barry J. (2001). *Case*. 2nd ed. Cambridge University Press.
- Bobaljik, Jonathan (2008). Where's Phi? Agreement as a Postsyntactic Operation. *Phi Theory*. Ed. by Daniel Harbour, David Adger, and Susana Béjar. Oxford: Oxford University Press, May 1, pp. 295–328. DOI: 10.1093/oso/9780199213764.003.0010.
- Boston, Marisa Ferrara, John T. Hale, and Marco Kuhlmann (2010). Dependency structures derived from Minimalist Grammars. *The Mathematics of Language. 10th and 11th Biennial Conference, MOL 10, Los Angeles, CA, USA, July 28-30, 2007 and MOL 11, Bielefeld,*

- Germany, August 20-21, 2009, Revised Selected Papers*. Ed. by Christian Ebert, Gerhard Jäger, and Jens Michaelis. Lecture Notes in Computer Science 6149. Berlin, Heidelberg: Springer, pp. 1–12. DOI: 10.1007/978-3-642-14322-9_1.
- Bouma, Gosse, Robert Malouf, and Ivan A. Sag (2001). Satisfying Constraints on Extraction and Adjunction. *Natural Language & Linguistic Theory* 19.1, pp. 1–65. DOI: 10.1023/a:1006473306778.
- Burness, Phillip Alexander et al. (2021). Long-distance phonological processes as tier-based strictly local functions. *Glossa: a journal of general linguistics* 6.1. DOI: 10.16995/glossa.5780.
- Butt, Miriam (2006). *Theories of case*. Cambridge University Press.
- Chandlee, Jane (2014). “Strictly local phonological processes”. PhD thesis. University of Delaware.
- Chandlee, Jane (2017). Computational locality in morphological maps. *Morphology* 27.4, pp. 599–641. DOI: 10.1007/s11525-017-9316-9.
- Chandlee, Jane and Jeffrey Heinz (2018). Strict Locality and Phonological Maps. *Linguistic Inquiry* 49.1, pp. 23–60. URL: <https://muse.jhu.edu/article/683698>.
- Chomsky, Noam (1956). Three models for the description of language. *IRE Transactions on information theory* 2.3, pp. 113–124. DOI: 10.1109/TIT.1956.1056813.
- Chomsky, Noam (1957). *Syntactic structures*. De Gruyter Mouton.
- Chomsky, Noam (1959). On certain formal properties of grammars. *Information and control* 2.2, pp. 137–167. DOI: 10.1016/S0019-9958(59)90362-6.
- Chomsky, Noam (1965). *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.
- Chomsky, Noam (1973). Conditions on Transformations. *A Festschrift for Morris Halle*. Ed. by Stephen R. Anderson and Paul Kiparsky. New York: Holt, Rinehart and Winston, pp. 232–286.
- Chomsky, Noam (1981). *Lectures on government and binding: The Pisa lectures*. Berlin, New York: Walter de Gruyter.
- Chomsky, Noam (1995). *The Minimalist Program*. Cambridge, MA: MIT Press. DOI: 10.7551/mitpress/9780262527347.001.0001.
- Chomsky, Noam (2000). Minimalist inquiries: The framework. *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*. Ed. by R. Martin, D. Michaels, and J. Uriagereka. Cambridge, MA: MIT Press, pp. 89–155. URL: <https://mitpress.mit.edu/9780262516839/step-by-step/>.

- Chomsky, Noam (2001). Derivation by Phase. *Ken Hale: A Life in Language*. Ed. by Michael Kenstowicz. Cambridge, MA: MIT Press, pp. 1–52.
- Chomsky, Noam (2008). On Phases. *Foundational Issues in Linguistic Theory: Essays in Honor of Jean-Roger Vergnaud*. Cambridge, MA: MIT Press, pp. 133–166.
- Clem, Emily and Amy Rose Deal (2024). Dependent Case by Agree: Ergative in Shawi. *Linguistic Inquiry*, pp. 1–47. DOI: 10.1162/ling_a_00529.
- Coon, Jessica and Clint Parker (2019). Case Interactions in Syntax. *Oxford Research Encyclopedia of Linguistics*. Oxford University Press, June. DOI: 10.1093/acrefore/9780199384655.013.315.
- Corbett, Greville G. (2006). *Agreement*. Cambridge University Press.
- Daniel, Penelope (2024). Unifying Differential Argument Marking through Interpretable Features. *Proceedings of the 41st West Coast Conference on Formal Linguistics*. Somerville, MA: Cascadilla Press, pp. 105–112.
- Davies, William D. and Stanley Dubinsky (2009). On the Existence (and Distribution) of Sentential Subjects. *Hypothesis A/Hypothesis B. Linguistic Explorations in Honor of David M. Perlmutter*. Ed. by Donna B. Gerdts, John C. Moore, and Maria Polinsky.
- De la Higuera, Colin (1997). Characteristic Sets for Polynomial Grammatical Inference. *Machine Learning* 27.2, pp. 125–138. DOI: 10.1023/a:1007353007695.
- De Santo, Aniello and Alëna Aksënova (2021). Learning Interactions of Local and Non-Local Phonotactic Constraints from Positive Input. *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*. Ed. by Allyson Ettinger, Ellie Pavlick, and Brandon Prickett. Amherst, MA: University of Massachusetts Amherst, pp. 167–176. DOI: 10.7275/M1AB-QV64.
- De Santo, Aniello and Thomas Graf (2019). Structure Sensitive Tier Projection: Applications and Formal Properties. *Formal Grammar: 24th International Conference*. FG 2019. Ed. by Raffaella Bernardi, Gregory Kobele, and Sylvain Pogodalla. Berlin, Heidelberg: Springer, pp. 35–50. DOI: 10.1007/978-3-662-59648-7_3.
- Deal, Amy Rose (2009). The origin and content of expletives: Evidence from “selection”. *Syntax* 12.4, pp. 285–323. DOI: 10.1111/j.1467-9612.2009.00127.x.
- Deal, Amy Rose (2015). Interaction and satisfaction in ϕ -agreement. *Proceedings of NELS 45*. Amherst, MA: GLSA Publications, pp. 179–192.
- Deal, Amy Rose (2024). Interaction, Satisfaction, and the PCC. *Linguistic Inquiry* 55.1, pp. 39–94. DOI: 10.1162/ling_a_00455.

- Diercks, Michael (2013). Indirect agree in Lubukusu complementizer agreement. *Natural Language & Linguistic Theory* 31.2, pp. 357–407. DOI: 10.1007/s11049-013-9187-7.
- Dolatian, Hossep and Jeffrey Heinz (2018). Modeling Reduplication with 2-way Finite-State Transducers. *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Ed. by Sandra Kuebler and Garrett Nicolai. Brussels, Belgium: Association for Computational Linguistics, pp. 66–77. DOI: 10.18653/v1/W18-5807.
- Dolatian, Hossep and Jeffrey Heinz (2020). Computing and classifying reduplication with 2-way finite-state transducers. *Journal of Language Modelling* 8.1, pp. 179–250. DOI: 10.15398/jlm.v8i1.245.
- Ermolaeva, Marina (2018). Morphological Agreement in Minimalist Grammars. *Formal Grammar: 22nd International Conference*. FG 2017. Ed. by Annie Foret, Reinhard Muskens, and Sylvain Pogodalla. Berlin, Heidelberg: Springer, pp. 20–36. DOI: 10.1007/978-3-662-56343-4_2.
- Ermolaeva, Marina and Gregory M. Kobele (2022). Agree as information transmission over dependencies. *Syntax* 25.4, pp. 466–507. DOI: 10.1111/synt.12240.
- Ermolaeva, Marina and Gregory M. Kobele (2023). Formal Properties of Agreeing Minimalist Grammars. *Proceedings of the Society for Computation in Linguistics (SCiL) 2023*. Amherst, MA: University of Massachusetts Amherst, pp. 385–388. DOI: 10.7275/KQY6-0737.
- Frey, Werner and Hans-Martin Gärtner (2002). On the treatment of scrambling and adjunction in minimalist grammars. *Proceedings of Formal Grammar 2002*. Ed. by G. Jäger et al., pp. 41–52.
- Fukuda, Shin (2007). Object Case and Event Type: Accusative-Dative Object Case Alternation in Japanese. *Annual Meeting of the Berkeley Linguistics Society*. Vol. 33. 1, pp. 165–176.
- Gleim, Daniel and Johannes Schneider (2023). Phonological processes with intersecting tier alphabets. *Proceedings of the Society for Computation in Linguistics (SCiL) 2023*. Amherst, MA: University of Massachusetts Amherst, pp. 243–249. DOI: 10.7275/6EMR-8380.
- Goldsmith, John (1976). “Autosegmental phonology”. PhD thesis. Cambridge, MA: Massachusetts Institute of Technology.
- Graf, Thomas (2013). “Local and Transderivational Constraints in Syntax and Semantics”. PhD thesis. University of California, Los Angeles. URL: <https://thomasgraf.net/output/graf13thesis.html>.
- Graf, Thomas (2017a). A Computational Guide to the Dichotomy of Features and Constraints. *Glossa: a journal of general linguistics* 2, pp. 1–36. DOI: 10.5334/gjgl.212.

- Graf, Thomas (2017b). The Power of Locality Domains in Phonology. *Phonology* 34, pp. 385–405. DOI: 10.1017/S0952675717000197.
- Graf, Thomas (2018). Why movement comes for free once you have adjunction. *Proceedings of CLS 53*. Chicago, IL: The Chicago Linguistic Society, pp. 117–136. URL: <https://thomasgraf.net/output/graf18cls.html>.
- Graf, Thomas (2019). Monotonicity as an effective theory of morphosyntactic variation. *Journal of Language Modelling* 7.2, pp. 3–47. DOI: 10.15398/jlm.v7i2.211.
- Graf, Thomas (2020a). Curbing Feature Coding: Strictly Local Feature Assignment. *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*. Amherst, MA: University of Massachusetts Amherst, pp. 224–233. DOI: 10.7275/f7y5-xz32.
- Graf, Thomas (2020b). Monotonicity in Syntax. *Monotonicity in Logic and Language*. Ed. by Dun Deng et al. Berlin, Heidelberg: Springer, pp. 35–53. DOI: 10.1007/978-3-662-62843-0_3.
- Graf, Thomas (2022a). Diving deeper into subregular syntax. *Theoretical Linguistics* 48.3–4, pp. 245–278. DOI: 10.1515/tl-2022-2043.
- Graf, Thomas (2022b). Subregular linguistics: bridging theoretical linguistics and formal grammar. *Theoretical Linguistics* 48.3–4, pp. 145–184. DOI: 10.1515/tl-2022-2037.
- Graf, Thomas (2022c). Typological implications of tier-based strictly local movement. *Proceedings of the Society for Computation in Linguistics (SCiL) 2022*. Amherst, MA: University of Massachusetts Amherst, pp. 184–193. DOI: 10.7275/gb65-ht31.
- Graf, Thomas (2023). Subregular Tree Transductions, Movement, Copies, Traces, and the Ban on Improper Movement. *Proceedings of the Society for Computation in Linguistics (SCiL) 2023*. Ed. by Tim Hunter and Brandon Prickett. Amherst, MA: University of Massachusetts Amherst, pp. 289–299. DOI: 10.7275/TK1N-Q855.
- Graf, Thomas (2024a). Minimalism and Computational Linguistics. *Cambridge Handbook of Minimalism*. Ed. by Kleanthes K. Grohmann and Evelina Leivada. Cambridge: Cambridge University Press. In press.
- Graf, Thomas (2024b). Morphologically simplex D-quantifiers are strictly 2-local. *Proceedings of the Society for Computation in Linguistics (SCiL) 2024*. University of Massachusetts Amherst Libraries, pp. 32–42. DOI: 10.7275/SCIL.2127.
- Graf, Thomas, Alëna Aksënova, and Aniello De Santo (2016). A Single Movement Normal Form for Minimalist Grammars. *Formal Grammar*. Ed. by Annie Foret et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 200–215.
- Graf, Thomas and Aniello De Santo (2019). Sensing Tree Automata as a Model of Syntactic

- Dependencies. *Proceedings of the 16th Meeting on the Mathematics of Language (MoL 16)*. Ed. by Philippe de Groote and Frank Drewes and Gerald Penn. Association for Computational Linguistics, pp. 12–26. DOI: 10.18653/v1/W19-5702.
- Graf, Thomas and Kenneth Hanson (2025). Automata for subregular syntax: Syntax with strings attached. *Proceedings of the Society for Computation in Linguistics (SCiL) 2025*. DOI: 10.7275/scil.3184.
- Graf, Thomas and Jeffrey Heinz (2015). *Commonality in Disparity: The Computational View of Syntax and Phonology*. Slides of a talk given at GLOW 2015, April 18, Paris, France.
- Graf, Thomas and Kalina Kostyszyn (2021). Multiple Wh-Movement is not Special: The Subregular Complexity of Persistent Features in Minimalist Grammars. *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*. Ed. by Allyson Ettinger, Ellie Pavlick, and Brandon Prickett. Amherst, MA: University of Massachusetts Amherst, pp. 275–285. DOI: 10.7275/2dfv-v219.
- Graf, Thomas and Connor Mayer (2018). Sanskrit n-Retroflexion is Input-Output Tier-Based Strictly Local. *Proceedings of SIGMORPHON 2018*. Association for Computational Linguistics, pp. 151–160. DOI: 10.18653/v1/W18-5817.
- Graf, Thomas and Nazila Shafiei (2019). C-command dependencies as TSL string constraints. *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*. Ed. by Gaja Jarosz et al. Amherst, MA: University of Massachusetts Amherst, pp. 205–215. DOI: 10.7275/4rrx-x488.
- Hanson, Kenneth (2023). A TSL Analysis of Japanese Case. *Proceedings of the Society for Computation in Linguistics (SCiL) 2023*. Ed. by Tim Hunter and Brandon Prickett. Amherst, MA: University of Massachusetts Amherst Libraries, pp. 15–24. DOI: 10.7275/xqhr-r404.
- Hanson, Kenneth (2024a). “A Tier-Based Analysis of Parasitic Agreement in Hindi-Urdu”. Oral presentation. SYNC 2024, New York University, New York, NY, Dec. 7.
- Hanson, Kenneth (2024b). A Tier-Based Model of Syntactic Agreement. *Proceedings of CLS 60*. Chicago, IL: The Chicago Linguistic Society. To appear.
- Hanson, Kenneth (2024c). Strict Locality in Syntax. *Proceedings of CLS 59*. Ed. by Kutay Serova and M. K. Snigaroff. Chicago, IL: The Chicago Linguistic Society, pp. 131–145.
- Hanson, Kenneth (2024d). Tiers, Paths, and Syntactic Locality: The View from Learning. *Proceedings of the Society for Computation in Linguistics (SCiL) 2024*. Ed. by Richard Futrell, Connor Mayer, and Noga Zaslavsky. Amherst, MA: University of Massachusetts Amherst Libraries, pp. 107–116. DOI: 10.7275/scil.2135.
- Hanson, Kenneth (2025a). Adjunction in (T)SL Syntax. *Proceedings of the Society for Computation in Linguistics (SCiL) 2025*. DOI: 10.7275/scil.3190.

- Hanson, Kenneth (2025b). The Typology of TSL Case Assignment. *Proceedings of the 18th Meeting on the Mathematics of Language*. To appear.
- Hanson, Kenneth (2025c). Tier-based strict locality and the typology of agreement. *Journal of Language Modelling* 13.1, pp. 43–97. DOI: 10.15398/jlm.v13i1.411.
- Hanson, Kenneth (2025d). “Two Steps to Parasitic Agreement in Hindi-Urdu”. Poster presentation. PLC 49, University of Pennsylvania, Philadelphia, PA, Apr. 5–6.
- Hansson, Gunnar Ólafur (2010). *Consonant Harmony: Long-Distance Interactions in Phonology*. University of California Press.
- Harley, Heidi and Elizabeth Ritter (2002). Person and Number in Pronouns: A Feature-Geometric Analysis. *Language* 78.3, pp. 482–526. URL: <http://www.jstor.org/stable/3086897>.
- Heinz, Jeffrey (2009). On the role of locality in learning stress patterns. *Phonology* 26.2, pp. 303–351. DOI: 10.1017/S0952675709990145.
- Heinz, Jeffrey (2010a). Learning Long-Distance Phonotactics. *Linguistic Inquiry* 41.4, pp. 623–661. DOI: 10.1162/LING_a_00015.
- Heinz, Jeffrey (2010b). String Extension Learning. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 897–906. URL: <https://dl.acm.org/doi/10.5555/1858681.1858773>.
- Heinz, Jeffrey (2014). Culminativity Times Harmony Equals Unbounded Stress. *Word Stress: Theoretical and Typological Issues*. Ed. by Harry van der Hulst. Cambridge, UK: Cambridge University Press. Chap. 8.
- Heinz, Jeffrey (2016). *Computational Theories of Learning and Developmental Psycholinguistics*. Ed. by Jeffrey L. Lidz, William Snyder, and Joe Pater. Oxford University Press, Nov. DOI: 10.1093/oxfordhb/9780199601264.013.27.
- Heinz, Jeffrey (2018). The computational nature of phonological generalizations. *Phonological Typology*. Ed. by Larry M. Hyman and Frans Plank. Phonetics and Phonology 23. De Gruyter Mouton, pp. 126–195. DOI: 10.1515/9783110451931-005.
- Heinz, Jeffrey and William Idsardi (2013). What Complexity Differences Reveal About Domains in Language. *Topics in Cognitive Science* 5.1, pp. 111–131. DOI: 10.1111/tops.12000.
- Heinz, Jeffrey, Chetan Rawal, and Herbert G. Tanner (2011). Tier-based strictly local constraints for phonology. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, pp. 58–64. URL: <https://aclanthology.org/P11-2011/>.

- Heuser, Annika, Hector Vazquez Martinez, and Charles Yang (2024). *The Learnability of Syntactic Islands*. Presentation at NELS 54, MIT.
- Himmelreich, Anke (2022). Locality in focus marking. *Theoretical Linguistics* 48.3-4, pp. 233–243.
- Hiraiwa, Ken (2001a). Multiple Agree and the Defective Intervention Constraint in Japanese. *Proceedings of the MIT-Harvard Joint Conference (HUMIT 2000)*. Cambridge, MA: MITWPL, pp. 67–80.
- Hiraiwa, Ken (2001b). On Nominative-Genitive Conversion. *MIT Working Papers in Linguistics* 39. Ed. by Elena Guerzoni and Ora Matushansky, pp. 66–125.
- Hirata, Ichiro (2006). Predicate coordination and clause structure in Japanese. *Linguistic Review* 23.1. DOI: 10.1515/TLR.2006.003.
- Holmberg, Anders and Thorbjörg Hróarsdóttir (2003). Agreement and movement in Icelandic raising constructions. *Lingua* 113.10, pp. 997–1019. DOI: 10.1016/S0024-3841(02)00162-6.
- Huybregts, Riny (1984). The weak inadequacy of context-free phrase structure grammars. *Van periferie naar kern*. Ed. by G. J. de Haan, M. Trommelen, and W. Zonneveld. Dordrecht: Foris, pp. 81–99.
- Ikawa, Shiori and Adam Jardine (2021). The computational similarity of binding and long-distance consonant dissimilation. *Proceedings of the 44th Annual Penn Linguistics Conference*. University of Pennsylvania Working Papers in Linguistics 27. URL: <https://repository.upenn.edu/pwpl/vol27/iss1/13/>.
- Ishizuka, Tomoko (2017). The passive voice. *Handbook of Japanese Syntax*. De Gruyter Mouton, pp. 403–446.
- Jardine, Adam (2016). Computationally, tone is different. *Phonology* 33.2, pp. 247–283. DOI: 10.1017/S0952675716000129.
- Jardine, Adam and Jeffrey Heinz (2016). Learning Tier-based Strictly 2-Local Languages. *Transactions of the Association for Computational Linguistics* 4, pp. 87–98. DOI: 10.1162/tac1_a_00085.
- Jardine, Adam and Kevin McMullin (2017). Efficient Learning of Tier-Based Strictly k-Local Languages. *Language and Automata Theory and Applications*. Ed. by Frank Drewes, Carlos Martín-Vide, and Bianca Truthe. Cham: Springer, pp. 64–76.
- Johnson, Jacob and Aniello De Santo (2024). Online Learning of ITSL Grammars. *Proceedings of the Society for Computation in Linguistics (SCiL) 2024*. University of Massachusetts Amherst Libraries, pp. 257–267. DOI: 10.7275/SCIL.2159.

- Joshi, Aravind K. (1985). Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*. Ed. by David R. Dowty, Lauri Karttunen, and Arnold M. Zwicky. Studies in Natural Language Processing. Cambridge University Press, pp. 206–250. DOI: 10.1017/CB09780511597855.007.
- Kalin, Laura (2020). Opacity in agreement. *Agree to Agree: Agreement in the Minimalist Programme*. Ed. by Peter W. Smith, Johannes Mursell, and Katharina Hartmann. Berlin: Language Science Press, pp. 149–177. DOI: 10.5281/ZENODO.3541751.
- Kaplan, Ronald M. and Martin Kay (1994). Regular models of phonological rule systems. *Computational linguistics* 20.3, pp. 331–378. URL: <http://www.aclweb.org/anthology/J94-3001.pdf>.
- Kayne, Richard S. (1984). *Connectedness and binary branching*. Berlin, Boston: De Gruyter. DOI: 10.1515/9783111682228.
- Keine, Stefan (2019). Selective Opacity. *Linguistic Inquiry* 50.1, pp. 13–62. DOI: 10.1162/ling_a_00299.
- Keine, Stefan (2020). *Probes and their horizons*. Linguistic Inquiry Monographs. Cambridge, MA: MIT Press.
- Kishimoto, Hideki (2018). On exceptional case marking phenomena in Japanese. *Kobe Papers in Linguistics* 11, pp. 31–49.
- Kobele, Gregory M. (2002). Formalizing mirror theory. *Grammars* 5.3, pp. 177–221. DOI: 10.1023/A:1022104104992.
- Kobele, Gregory M. (2006). “Generating copies: An investigation into structural identity in lan-guage and grammar”. PhD thesis. University of California, Los Angeles.
- Kobele, Gregory M. (2012). Eliding the derivation: A minimalist formalization of ellipsis. *Proceedings of the International Conference on Head-Driven Phrase Structure Grammar*. CSLI Publications, Oct. DOI: 10.21248/hpsg.2012.23.
- Kroch, Anthony S. (1989). Reflexes of grammar in patterns of language change. *Language Variation and Change* 1.03, pp. 199–244. DOI: 10.1017/S0954394500000168.
- Kučerová, Ivona (2016). Long-Distance Agreement in Icelandic: locality restored. *The Journal of Comparative Germanic Linguistics* 19.1, pp. 49–74. DOI: 10.1007/s10828-016-9077-6.
- Kuno, Susumu (1973). *The Structure of the Japanese Language*. Cambridge, MA: MIT Press.
- Lambert, Dakotah (2021). Grammar Interpretations and Learning TSL Online. *Proceedings of the Fifteenth International Conference on Grammatical Inference*. Ed. by Jane Chandlee

- et al. Vol. 153. Proceedings of Machine Learning Research. PMLR, pp. 81–91. URL: <https://proceedings.mlr.press/v153/lambert21a.html>.
- Lambert, Dakotah (2022). “Unifying Classification Schemes for Languages and Processes With Attention to Locality and Relativizations Thereof”. PhD thesis. Stony Brook University.
- Lambert, Dakotah (2023). Relativized Adjacency. *Journal of Logic, Language and Information* 32.4, pp. 707–731. DOI: 10.1007/s10849-023-09398-x.
- Lambert, Dakotah (2025). Multitier Phonotactics. *Phonology*. To appear.
- Lambert, Dakotah, Jonathan Rawski, and Jeffrey Heinz (2021). Typology emerges from simplicity in representations and learning. *Journal of Language Modelling* 9.1, pp. 151–194. DOI: 10.15398/jlm.v9i1.262.
- Lambert, Dakotah and James Rogers (2020). Tier-Based Strictly Local Stringsets: Perspectives from Model and Automata Theory. *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*. Ed. by Allyson Ettinger, Gaja Jarosz, and Joe Pater. Amherst, MA: University of Massachusetts Amherst, pp. 159–166. DOI: 10.7275/2n1j-pj39.
- Laszakovits, Sabine (2018). Case Theory in Minimalist Grammars. *Formal Grammar 2018: 23rd International Conference*. FG 2018. Ed. by Annie Foret, Greg Kobele, and Sylvain Pogodalla. Berlin, Heidelberg: Springer, pp. 37–61. DOI: 10.1007/978-3-662-57784-4_3.
- Legate, Julie Anne (2008). Morphological and abstract case. *Linguistic Inquiry* 39.1, pp. 55–101. DOI: 10.1162/ling.2008.39.1.55.
- Levin, Theodore and Omer Preminger (2015). Case in Sakha: Are two modalities really necessary? *Natural Language & Linguistic Theory* 33, pp. 231–250. DOI: 10.1007/s11049-014-9250-z.
- Lohndal, Terje (2014). Sentential Subjects: Topics or Real Subjects? *Proceedings of the 31st West Coast Conference on Formal Linguistics*. Cascadilla Proceedings Project, pp. 315–324.
- Lohninger, Magdalena, Iva Kovač, and Susanne Wurmbrand (2022). From Prolepsis to Hyper-raising. *Philosophies* 7.2. DOI: 10.3390/philosophies7020032.
- Mahajan, Anoop Kumar (1990). “The A/A-bar distinction and movement theory”. PhD thesis. Cambridge, MA: Massachusetts Institute of Technology.
- Maki, Hideki and Asako Uchibori (2008). Ga/no conversion. *The Oxford Handbook of Japanese Linguistics*.

- Malchukov, Andrej and Andrew Spencer, eds. (2008). *The handbook of case*. Oxford: Oxford University Press.
- Marantz, Alec (2000). Case and licensing. *Arguments and Case: Explaining Burzio's generalization*, pp. 11–30.
- Markowska, Magdalena and Jeffrey Heinz (2023). Empirical and Theoretical Arguments for Using Properties of Letters for the Learning of Sequential Functions. *Proceedings of 16th edition of the International Conference on Grammatical Inference*. Ed. by François Coste, Faissal Ouardi, and Guillaume Rabusseau. Vol. 217. Proceedings of Machine Learning Research. PMLR, pp. 270–274. URL: <https://proceedings.mlr.press/v217/markowska23a.html>.
- Markowska, Magdalena and Jeffrey Heinz (2025). “Learning k -ISL functions with phonological features”. Ms., Stony Brook University.
- Mayer, Connor (2021). Capturing gradience in long-distance phonology using probabilistic tier-based strictly local grammars. *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*. Ed. by Allyson Ettinger, Ellie Pavlick, and Brandon Prickett. Amherst, MA: University of Massachusetts Amherst, pp. 39–50. DOI: 10.7275/231q-p480.
- Mayer, Connor and Travis Major (2018). A challenge for tier-based strict locality from Uyghur backness harmony. *Formal Grammar 2018: 23rd International Conference*. FG 2018. Berlin, Heidelberg: Springer, pp. 62–83. DOI: 10.1007/978-3-662-57784-4_4.
- McCollum, Adam G. et al. (2020). Unbounded circumambient patterns in segmental phonology. *Phonology* 37.2, pp. 215–255. DOI: 10.1017/S095267572000010X.
- McFadden, Thomas (2004). “The position of morphological case in the derivation: A study on the syntax-morphology interface”. PhD thesis. University of Pennsylvania.
- McFadden, Thomas, Sandhya Sundaresan, and Hedde Zeijlstra (2019). *Deriving selective opacity in adjuncts*. Poster presentation at GLOW 42, May 8–10, University of Oslo.
- McMullin, Kevin (2016). “Tier-based locality in long-distance phonotactics: learnability and typology”. PhD thesis. Vancouver, B.C.: University of British Columbia.
- McMullin, Kevin, Alëna Aksënova, and Aniello De Santo (2019). Learning phonotactic restrictions on multiple tiers. *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*. Ed. by Gaja Jarosz et al. Amherst, MA: University of Massachusetts Amherst, pp. 377–378. DOI: 10.7275/s8ym-bx57.
- McMullin, Kevin and Gunnar Ólafur Hansson (2016). Long-Distance Phonotactics as Tier-Based Strictly 2-Local Languages. *Proceedings of the Annual Meetings on Phonology*. Ed. by Adam Albright and Michelle A. Fullwood. Vol. 2. Washington, DC: Linguistic Society of America. DOI: 10.3765/amp.v2i0.3750.

- McNaughton, Robert and Seymour A. Papert (1971). *Counter-Free Automata*. MIT Press.
- Mikkelsen, Line (2005). *Copular clauses: Specification, predication and equation*. Vol. 85. *Linguistics Today*. John Benjamins.
- Miyagawa, Shigeru (1989). *Structure and Case Marking in Japanese*. Academic Press.
- Miyagawa, Shigeru and Takae Tsujioka (2004). Argument structure and ditransitive verbs in Japanese. *Journal of East Asian Linguistics* 13.1, pp. 1–38. DOI: 10.1023/B:JEAL.0000007345.64336.84.
- Nevins, Andrew (2007). The representation of third person and its consequences for person-case effects. *Natural Language & Linguistic Theory* 25.2, pp. 273–313. DOI: 10.1007/s11049-006-9017-2.
- Nevins, Andrew (2011). Multiple agree with clitics: Person complementarity vs. omnivorous number. *Natural Language & Linguistic Theory* 29, pp. 939–971. DOI: 10.1007/s11049-011-9150-4.
- Ozaki, Satoru (2014). *Dependent Case in Japanese*. Presentation at the 14th Annual Cornell Undergraduate Linguistics Colloquium (CULC14).
- Pesetsky, David and Esther Torrego (2001). T-to-C Movement: Causes and Consequences. *Ken Hale: A Life on Language*. Cambridge, MA: MIT Press.
- Pesetsky, David and Esther Torrego (2007). The Syntax of Valuation and the Interpretability of Features. *Phrasal and Clausal Architecture: Syntactic Derivation and Interpretation*. Ed. by Simin Karimi, Vida Samiian, and Wendy K. Wilkins. Amsterdam: Benjamins. DOI: 10.1075/1a.101.14pes.
- Polinsky, Maria and Eric Potsdam (2001). Long-distance agreement and topic in Tsez. *Natural Language & Linguistic Theory* 19.3, pp. 583–646. DOI: 10.1023/A:1010757806504.
- Poole, Ethan (2022). Improper case. *Natural Language & Linguistic Theory* 41.1, pp. 347–397. DOI: 10.1007/s11049-022-09541-6.
- Poole, Ethan (2024). Dependent-case assignment could be AGREE. *Glossa: a journal of general linguistics* 9.1. DOI: 10.16995/glossa.9894.
- Preminger, Omer (2014). *Agreement and its failures*. Vol. 68. *Linguistic Inquiry Monographs*. Cambridge, MA: MIT Press.
- Preminger, Omer (2024). Taxonomies of case and ontologies of case. *The Place of Case in Grammar*. Oxford, UK: Oxford University Press, July, pp. 73–92. DOI: 10.1093/oso/9780198865926.003.0004.

- Reinhart, Tanya (1981). Definite NP anaphora and c-command domains. *Linguistic Inquiry* 12.4, pp. 605–635. URL: <http://www.jstor.org/stable/4178245>.
- Rezac, Milan, Pablo Albizu, and Ricardo Etxepare (2014). The structural ergative of Basque and the theory of Case. *Natural Language & Linguistic Theory* 32.4, pp. 1273–1330. DOI: 10.1007/s11049-014-9239-7.
- Rizzi, Luigi (1990). *Relativized minimality*. Linguistic Inquiry Monographs 16. Cambridge, MA: MIT Press.
- Rizzi, Luigi (2001). Relativized Minimality Effects. *The Handbook of Contemporary Syntactic Theory*. John Wiley & Sons, Ltd. Chap. 4, pp. 89–110. DOI: 10.1002/9780470756416.ch4.
- Rizzi, Luigi (2013). Locality. *Lingua* 130, pp. 169–186. DOI: 10.1016/j.lingua.2012.12.002.
- Roark, Brian and Richard Sproat (2007). *Computational approaches to morphology and syntax*. Oxford: Oxford University Press.
- Rogers, James (1997). Strict LT_2 : Regular :: Local : Recognizable. *Logical Aspects of Computational Linguistics: First International Conference, LACL '96 (Selected Papers)*. Ed. by Christian Retoré. Vol. 1328. Lectures Notes in Computer Science/Lectures Notes in Artificial Intelligence. Springer, pp. 366–385. DOI: 10.1007/BFb0052167.
- Rogers, James (1998). *Descriptive Approach to Language-Theoretic Complexity*. CSLI Publications.
- Rogers, James and Dakotah Lambert (2019a). Extracting Subregular constraints from Regular stringsets. *Journal of Language Modelling* 7.2. DOI: 10.15398/jlm.v7i2.209.
- Rogers, James and Dakotah Lambert (2019b). Some classes of sets of structures definable without quantifiers. *Proceedings of the 16th Meeting on the Mathematics of Language (MoL 16)*. Ed. by Philippe de Groote and Frank Drewes an Gerald Penn. Association for Computational Linguistics, July, pp. 63–77. DOI: 10.18653/v1/W19-5706.
- Rogers, James and Geoffrey K. Pullum (2011). Aural Pattern Recognition Experiments and the Subregular Hierarchy. *Journal of Logic, Language and Information* 20.3, pp. 329–342. DOI: 10.1007/s10849-011-9140-2.
- Rogers, James et al. (2010). On Languages Piecewise Testable in the Strict Sense. *The Mathematics of Language. 10th and 11th Biennial Conference, MOL 10, Los Angeles, CA, USA, July 28-30, 2007 and MOL 11, Bielefeld, Germany, August 20-21, 2009, Revised Selected Papers*. Ed. by Christian Ebert, Gerhard Jäger, and Jens Michaelis. Lecture Notes in Computer Science 6149. Berlin, Heidelberg: Springer, pp. 255–265. DOI: 10.1007/978-3-642-14322-9_19.

- Rogers, James et al. (2013). Cognitive and Sub-Regular Complexity. *Formal Grammar*. Ed. by Glyn Morrill and Mark-Jan Nederhof. Vol. 8036. Lecture Notes in Computer Science. Springer, pp. 90–108. DOI: 10.1007/978-3-642-39998-5_6.
- Rose, Sharon and Rachel Walker (2004). A Typology of Consonant Agreement as Correspondence. *Language* 80.3, pp. 475–531. DOI: 10.1353/lan.2004.0144.
- Ross, John Robert (1967). “Constraints on variables in syntax”. PhD thesis. Cambridge, MA: Massachusetts Institute of Technology.
- Saito, Mamoru (2015). Cartography and Selection: Case Studies in Japanese. *Beyond Functional Sequence*. Oxford University Press, May, pp. 255–274. DOI: 10.1093/acprof:oso/9780190210588.003.0014.
- Saito, Mamoru, T.-H. Jonah Lin, and Keiko Murasugi (2008). N’-Ellipsis and the Structure of Noun Phrases in Chinese and Japanese. *Journal of East Asian Linguistics* 17.3, pp. 247–271. DOI: 10.1007/s10831-008-9026-8.
- Shafiei, Nazila and Thomas Graf (2020). The Subregular Complexity of Syntactic Islands. *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*. Ed. by Allyson Ettinger, Gaja Jarosz, and Joe Pater. Amherst, MA: University of Massachusetts Amherst, pp. 421–430. DOI: 10.7275/scil.1202.
- Shieber, Stuart M. (1985). Evidence against the context-freeness of natural language. *The formal complexity of natural language*. Springer, pp. 320–334. URL: <http://dx.doi.org/10.1007/BF00630917>.
- Sigurðsson, Halldór Ármann and Anders Holmberg (2008). Icelandic dative intervention: Person and number are separate probes. *Agreement restrictions*. Ed. by Roberta D’Alessandro, Susann Fischer, and Gunna Hrafn Hrafnbjargarson. Berlin: De Gruyter Mouton, pp. 176–194.
- Stabler, Edward P. (1997). Derivational minimalism. *Logical Aspects of Computational Linguistics. First International Conference, LACL ’96, Nancy, France, September 23-25, 1996. Selected Papers*. Ed. by Christian Retoré. Lecture Notes in Computer Science 1328. Berlin, Heidelberg: Springer. DOI: 10.1007/BFb0052152.
- Stabler, Edward P. (2011). Computational perspectives on Minimalism. *Oxford handbook of linguistic Minimalism*. Ed. by Cedric Boeckx. Oxford: Oxford University Press, pp. 617–643. DOI: 10.1093/oxfordhb/9780199549368.013.0027.
- Swanson, Logan (2024). Syntactic Learning over Tree Tiers. *Proceedings of ESSLLI 2024 Student Session*. 35th European Summer School in Logic, Language, and Information. Ed. by Francesco Di Cosmo and John Lindqvist, pp. 187–196.
- Thrainsson, Hoskuldur (2007). *The syntax of Icelandic*. Cambridge University Press.

- Torr, John (2017). Autobank: a semi-automatic annotation tool for developing deep Minimalist Grammar treebanks. *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain: Association for Computational Linguistics, Apr., pp. 81–86. URL: <https://aclanthology.org/E17-3021>.
- Torres, Charles et al. (2023). Modeling island effects with probabilistic tier-based strictly local grammars over trees. *Proceedings of the Society for Computation in Linguistics (SCiL) 2023*. Ed. by Tim Hunter and Brandon Prickett. Amherst, MA: University of Massachusetts Amherst Libraries, pp. 155–164. DOI: 10.7275/nz4q-6b09.
- Uriagereka, Juan (1999). Multiple Spell-Out. *Working minimalism*. Ed. by Samuel D. Epstein and Norbert Hornstein. Cambridge, MA: MIT Press.
- Van Urk, Coppe (2015). “A Uniform Syntax for Phrasal Movement: A Dinka Bor Case Study”. PhD thesis. Cambridge, MA: Massachusetts Institute of Technology.
- Vergnaud, Jean-Roger (1977). “Letter to Noam Chomsky and Howard Lasnik”.
- Vu, Mai Ha, Nazila Shafiei, and Thomas Graf (2019). Case assignment in TSL syntax: A case study. *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*. Ed. by Gaja Jarosz et al. Amherst, MA: University of Massachusetts Amherst, pp. 267–276. DOI: 10.7275/sy wz-xw23.
- Witzlack-Makarevich, Alena and Ilja A. Seržant (2018). Differential argument marking: Patterns of variation. *Diachrony of differential argument marking*. Ed. by Ilja A. Seržant and Alena Witzlack-Makarevich. Studies in Diversity Linguistics 19. Berlin: Language Science Press. DOI: 10.5281/zenodo.1219168.
- Yang, Charles (2016). *The price of linguistic productivity: How children learn to break the rules of language*. Cambridge, MA: MIT Press.
- Yip, Moira, Joan Maling, and Ray Jackendoff (1987). Case in Tiers. *Language* 63.2, pp. 217–250. URL: <http://www.jstor.org/stable/415655>.
- Zaenen, A., J. Maling, and H. Thráinsson (1985). Case and Grammatical Functions: The Icelandic Passive. *Natural Language & Linguistic Theory* 3.4, pp. 441–483. URL: <http://www.jstor.org/stable/4047609>.
- Zeijlstra, Hedde (2012). There is only one way to agree. *The Linguistic Review* 29.3, pp. 491–539. DOI: 10.1515/tlr-2012-0017.