Computational Locality and Domain of Syntactic Long-distance Dependencies

A Dissertation Presented

by

Nazila Shafiei

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Linguistics

Stony Brook University

May 2022

Stony Brook University

The Graduate School

Nazila Shafiei

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 Frederick Bailyn
Professor, Department of Linguistics

Dennis Ryan Storoshenko
Associate Professor, School of Languages, Linguistics, Literatures and Cultures,
University of Calgary

This dissertation is accepted by the Graduate School

Eric Wertheimer
Dean of the Graduate School

Abstract of the Dissertation

Computational Locality and Domain of Syntactic Long-distance Dependencies

by

Nazila Shafiei

Doctor of Philosophy

in

Linguistics

Stony Brook University

2022

This thesis investigates syntactic long-distance dependencies from a computational viewpoint. The two main phenomena explored are *binding conditions* and *island constraints*. A formal framework for Minimalist Syntax, namely Minimalist grammars (MGs) (Stabler, 2011) is used as a means to enable formal examination of these phenomena. An intermediate representation is defined and utilized to display core relations in these conditions. These relations are c-command and dominance.This representation is obtained from MG dependency trees, and it encodes dependencies in a string format, rather than a tree format. For binding, this string format, named *c-strings*, involves c-command relations between different lexical items. A similar representation, *a-strings*, is used for island constraints, with a difference that it encodes the ancestry or dominance relation between different lexical items on an MG dependency tree.

Having the dependencies represented in a string form, I analyze these dependencies from a computational perspective, more specifically, from a Formal Language Theory perspective. I first show that having a computational perspective, each of these phenomena form a unified class and belong to one formal language class. Moreover, I show that using such representation(s), these phenomena are computationally much less complex than

generally assumed because they can be modeled in the *subregular* region of the extended Chomsky hierarchy over strings (Chomsky, 1956). A third nice corollary of my findings is that syntax shows parallels with phonology and morphology, which have been proposed to also fit in the *subregular* classes of string languages (see Heinz, 2010; Chandlee, 2014; a.o.). Determining the complexity of such operations has resulted in defining new learning algorithms, which have implications in the field of acquisition and how humans generalize patterns (see Heinz et al., 2015 and references therein).

Erdem'e

Contents

List of Figures

List of Tables

Acknowledgements

Every success story has a protagonist. In reality, it takes a group of people that come together to make that one person the heroine of that story. My success story is no different. I was lucky enough to come across a group of intelligent and awesome people who pushed me to better myself, go out of my comfort zone and achieve a lifetime goal of holding a Ph.D. degree. They were my academic and mental support, whom I cherish and admire. With the hope of not forgetting any name, I would like to express my gratitude to this wonderful group of people.

First and foremost, I would like to thank my advisor and my mentor, Thomas Graf, without whose help I would not be able to have come this far. He taught me to dream big, and always reminded me of the power of perseverance in order to succeed. His knowledge of the field as well as his responsible character made him an ideal advisor for me, someone with whom I could share my concerns and from whom I could learn a lot. Thomas, I am greatly indebted to you. Thanks for patiently guiding me and answering my never-ending questions.

I had many doubts when I switched from theoretical linguistics to computational linguistics, but I met Jeffrey Heinz with his passion for math and a never-fading smile on his face. His encouraging and positive character was a blessing that came into my then fragile academic life. Every time I left his classes, I would tell myself: “I can do it”. I am forever grateful to you, Jeff.

I cannot even begin to thank John Bailyn. People who know him, know how smart and funny and approachable he is. I always felt that his brain waves and my brain waves were on the same frequency, when it came to syntax of course! More often, he would mention things that I was thinking about seconds ago. It was so surreal and amazing. I will never forget our discussions in his office and his writings on his office blackboard. Thank you, John for being part of my academic journey. Keep being awesome.

To Dennis Storoshenko, whose support for me spans for almost a decade, starting in 2014 when I began my Master's studies in Calgary and continuing until my defense in February 2022. Thank you so very much for your interest in my academic life and for your support as an advisor before and as a committee member now.

Every Ph.D. student would know that without peer support, it is almost impossible to go through such an overwhelming experience. I am forever indebted to my fellows in this matter as well. I started my Ph.D. journey in 2016, and soon after, I connected with Aniello De Santo and Alëna Aksënova from the 2015 cohort. Setting aside all the crazy partying we did together, we had so many great linguistic and career discussions. I was and still am inspired by the two of you, and for that, I want to thank you both.

I will never forget our secret Signal group with Hossep Dolatian, Mai Ha Vu and Sabine Laszakovits, where we not only vented, but also had a lot of lively discussions about life choices and linguistics. We talked about our career plans, our ambitions and we shared about our lives. We also discussed papers like Brody 2015, which to this day I am not sure I fully understand (oops, did I say much? Afterall, I have cited that paper in the current dissertation!). I am so grateful to you for all your help and support; for Hossep's taking editorial role and helping me with revisions, for Mai Ha and Sabine's help with syntax, analysis and my final presentation, and for all of your being there just a call away whenever I needed you. I wish I could return the favor someday, and I hope we get to experience going to conferences like the crazy New Orlean's LSA conference Hossep, which is one of the most memorable conferences I have ever been to.

Special thanks to my Persian friends, Vahideh Rasekhi and Neda Taherkhani for providing mental support to me by helping me stay positive. Vahideh, having a fellow Tabrizi in the department, with whom I had so much in common was awesome. Your interpersonal skills have helped me on more than one occasion. And Neda, you always know what to tell me to make me feel calm. Thank you both so very much.

My parents also deserve special thanks, whose emotional support never ceased. Although they never fully understood why I had to be a student for such a long time and what Ph.D. life looks like, they have always been proud of me for pursuing my dreams and for becoming the first college graduate and first Ph.D. holder in the family.

Lastly, I owe a great deal to my husband, Erdem Gündoğdu, who has always been by my side during this journey and who has been so understanding and caring, who wiped my tears when my energy drained and who gave me the encouragement when I even stopped believing in myself. More special thanks for the past two years when I needed to manage a full-time job and writing a dissertation and took away almost every weekend from him. You will always have a special place in my heart.

Thank you all.

Chapter 1

Introduction

This thesis explores the computational complexity of various syntactic long-distance dependencies in an attempt to provide a unified account of them. This viewpoint is rooted in theoretical findings which provide a solid ground and diverse angles on these operations. The theoretical works also present a great source of empirical data both in English and cross-linguistically. What is missing in theoretical syntax is a unified account of such phenomena, regardless of all the attempts that have been made to classify the data under one group. I provide a new and fresh computational perspective on two extensively discussed phenomena, namely *binding conditions* and *island constraints*. I show that using such a viewpoint, each of these phenomena forms a natual class, as each can be unified under a formal class in the extended version of the Chomsky hierarchy (Chomsky, 1956).

Syntactic dependencies can be categorized into two main ones, namely *licensing* conditions and *constraints* on movement. To be more specific, while some dependencies require the existence of a specific *licensor*, such as dependency between a reflexive and its licensing antecedent, other dependencies are *constrained* by specific elements or phrases.

To elaborate, let's take the following examples for *reflexive binding* in (1), and compare them to a constraint on movement, such as the *Complex NP Constraint* (CNPC) in (2b).

(1) a. John shaved himself.
    b. * Mary shaved himself.

(2) a. Who does he believe [$_{\text{CP}}$ John saw _]?
    b. * Who does he believe [$_{\text{DP}}$ the claim [$_{\text{CP}}$ that John saw _]]?

Examples in (1) show, what I call, a *licensing* dependency. This is because the presence of the reflexive is only allowed when it is licensed by a potential antecedent. This is a requirement for the well-formedness of the sentence.

In the case of *wh*-movement, on the other hand, what we are observing is some sort of an intervention or blocking effect between the *wh*-mover and its landing site. While the dependency is uninterrupted in the first example in (2a), the presence of complex DP in (2b) blocks this dependency. In other words, the movement is constrained by such an element.

Given that these two cases differ slightly in nature, the prediction is that they can be modeled in different classes, and we will see in this dissertation that this holds. This does not mean that they do not have overlaps or that each can only fit in one class and not the other. Rather, the language classes I propose for them seem to be a natural fit. These two phenomena each have separate established empirical properties and theoretical analyses. Based on these properties and analyses, I develop a computational analysis in terms of Formal Language Theory (FLT), which is a means to assess the generative capacity of sets of objects generated by a grammar. I use Minimalist grammars (MGs) §2.1, which are based on the Minimalist Syntax, to represent the syntactic structures. Furthermore, I use MG dependency trees to obtain the relations I care about directly and to represent them in a string format. I propose the *subregular* class of IO-TSL (Input-Output Tier-based Strictly Local) as an upper bound that unifies binding relations, while island constraints fit natually into another *subregular* class, namely the IBSP (Interval-Based Strictly Piecewise). Not only does this result unifies these operations, it also shows that syntactic dependencies are in parallel with phonological and morpholocial dependenices in terms of their generative capacity.

This, in turn, will have serious implications in learnability considerations. Certain *subregular* classes over strings are shown to be learnable in the limit from positive data (Heinz, 2010; Jardine and Heinz, 2016; De Santo and Graf, 2019; a.o.). Therefore, establishing the *subregularity* of syntax can result in new learning algorithms, which can inform our understanding of acquisition of such patterns, as well as give ways to new Natural Language Processing (NLP) and Machine Learning (ML) techniques (see Aksënova (2020)'s language modeling experiments as an example of how the implementation of these algorithms results in learning phonotactic patterns).

While this thesis provides in-depth analysis of *reflexive binding*, and sets up the ground for studies of other binding conditions, it does not go into details of conditions B and C of binding. The focus is on Principle A of binding with some pointers for future work on Principle B and Principle C.

1.1 String-based Representation

It is well-known that syntactic dependencies are not regular over strings, i.e. sentences or yields of trees. Examples like (3) for English center-embedding take syntax out of the realm of *regular* and to the world of *context-free* languages (Chomsky, 1956).

(3) The rat the cat the dog chased killed ate the malt.

There are other phenomena, such as Swiss-German cross serial verb constructions, as (4) shows, that make syntax even more complex. Such constructions move syntax into a more powerful class of *mildly context-sensitive*, or as Shieber (1985) puts it *weakly non-context-free*.

(4)	... mer	d'chind	em Hans	es huus	lönd	hälfe	aastriiche	
	... we	the children-ACC	Hans-DAT	the house-ACC	let	help	paint	
	... we let the children help Hans paint the house							(Shieber, 1985: ex.5)

Recent works in the area of subregular hierarchy focus on the differences between various representation types, showing that choosing the right data structure can draw parallels between different subfields of linguistics (Graf, 2018a,b; Graf et al., 2018). For instance, using string-based representations for phonology, Graf (2018a) provides a subregular account of four different and unrelated phonological phenomena, namely Korean vowel harmony, the non-final *Rightmost Heavy, Otherwise Leftmost* (RHOL) stress pattern, non-local blocking of local dissimilation in Samala, and non-locally conditioned local tone spreading in Copperbelt Bemba. On the other hand, Graf et al. (2018) show that some local syntactic dependencies like *Merge* and some non-local dependencies like *Move* are also subregular over trees, in parallel with local and non-local phonological dependencies over strings.

The question that might come to mind is why we should use string-based representations, given that syntax is normally represented by trees. Heinz (2018) discusses this point for the study of phonological operations mentioning the prominence of string-based representations as one reason for such a choice. The second reason he indicates is the fact that strings are well-studied, being the fundamental data structures. Moreover, the findings for strings can be extended to other types of representations. In a nutshell, "if we want to understand how computational principles play out with complicated data structures, we better first understand how they play out with simpler structures like strings (Heinz, 2018: 186)".

However, while it is shown that the syntactic operations of *Merge* and *Move* are subregular over trees (Graf, 2018b), it is not clear if other types of operations, in particular c-command dependencies or movement constraints, are subregular over tree representations. Given the findings in phonology and morphology and the use of strings for such findings, it seems that if we can find a way to represent syntax with strings, we can keep it within the class of subregular languages. This is the idea that was presented and applied in Graf and Shafiei (2019) for the first time to deal with c-command dependencies, and has since been pursued by other scholars in various other works.

Graf and Shafiei (2019) use a mechanism (§2.2) to convert the dependencies on syntactic trees to strings. Using these strings and following the definitions of subclasses subsumed within the class of *regular* languages, they show that the c-command dependencies can in fact fit into the subregular class of TSL (Tier-based Strictly Local). This means that these dependencies are even less complex than the class of *regular* languages. Laszakovits and Graf (2020) use this mechanism of converting tree dependencies to path constraints to diagnose movement, showing that movement dependencies such as parastic gaps and Across The Board (ATB) movement cannot be accounted for using string-based representations. On the other hand, though, Shafiei and Graf (2020) show that island constraints can also be accounted for within the subregular class of IBSP. It is safe to conclude that representation matters.

This thesis presents a more comprehensive look at the use of string-based representations as an intermediate representation, focusing on its advantages and discussing its shortcomings and ideas for future work and directions. This strategy is evaluated in various dependency types, more specifically in licensing-type dependencies, such as binding relations, discussed in Chapter 3 versus constraint-type dependencies, such as island constraints, discussed in Chapter 4.

1.2 Command-strings versus Ancestor-strings

The two string representations that are used in this thesis are c[ommand]-strings (*c-strings*) and a[ncestor]-strings (*a-strings*). The former encodes c-command relations between various nodes or lexical items in an MG dependency tree, while the latter encodes ancestors of a given node, again on an MG dependency tree. I use the first one to analyze binding relations and the second one to analyze island constraints.

1.2.1 Command-strings for Binding

The notion of c(onstituent)-command dates back to Reinhart (1976) as defined in (5). A more modified definition to conform to more updated linguistic frameworks is provided in (6). At its core definition, this notion has been used to account for a variety of syntactic phenomena, including binding, NPI licensing, weak cross-over, movement, agreement, and other syntactic operations. For instance, binding happens through c-command and co-indexation; NPIs must be c-commanded by a negative element, and a probe-goal relation is established via c-command (Hornstein et al., 2005). C-command is also used as a diagnosing tool to derive English double object constructions, where the dative must c-command the theme, in Larson (1988).

(5) A node A c(onstituent)-commands node B iff the first branching node α_1 dominating A either dominates B or is immediately dominated by a node α_2 which dominates B, and α_2 is of the same category type as α_1. (Reinhart, 1976: 148)

(6) α c-commands β iff every node properly dominating α dominates β and neither dominates the other. (Frank and Vijay-Shanker, 2001: 168)

C-command, as it is defined, can be derived from dominance. However, Frank and Vijay-Shanker (2001) propose this syntactic relation to be the primitive relation instead of dominance, arguing that it is dominance that should be derived from c-command. They base their arguments on the fact that compared to dominance, using c-command, one can restrict the range of possible syntactic structures in a linguistically more natural way. Another prominent work suggesting the primitiveness of c-command is that of Epstein et al. (1998). They argue that this relation falls naturally from the operations of *Merge* and *Move*, which makes c-command "the only syntactic relation that is natural without further stipulation (Epstein et al., 1998: 10)" (see Bruening (2014) for a challenging view).

Given the significance of this relation in the syntactic literature, a great deal of syntactic dependencies are explainable using c-command. My thesis provides a new perspective on the computational complexity of binding dependencies by using c-command as the primitive relation. I define a mechanism that encodes c-command relations directly from MG dependency trees into a string representation, which I call *c-strings*. Although dominance and c-command can be derived from each other, it is shown that using c-command as the primitive relation, binding relations can be grouped together under one single formal language class of IO-TSL (Input-Output Tier-based Strictly Local), defined in §3.4.2. This means that the computational complexity of such a dependency is significantly reduced from the proposed *mildly context-sensitive* class, which makes them comparable to phonological and morphological dependencies.

1.2.2 Ancestor-strings for Islands

Islands involve an illegitimate movement operation from a base position to a c-commanding landing site. In other words, the notion of c-command is still relevant for island constraints. However, given that movement can only happen to a c-commanding position, using *c-strings* to represent c-commanding nodes for a mover is not necessary. In other words, we are not trying to enforce c-command relation between two nodes, rather we are trying to determine whether a movement is licit or not. As a result, I use a different representation to analyze island constraints. This representation, which I call *a-strings*, lists ancestor relations directly from MG dependency tree into a string-based format. This relation encodes a mother-daughter relation between a mover and its landing site, which is the relation required for my analysis.

Similar to the binding conditions, using this string-based representation, I show that island constraints fit into the IBSP class over strings and form a unified group. The IBSP (Interval-Based Strictly Piecewise) class is a *subregular* class, defined in §4.3. Similar to the binding conditions, this shows that the complexity of these constraints is comparable to phonological and morphological dependencies.

Another major finding is that by using *a-strings*, I show that other movement constraints, such as the *that*-trace effect and Coordinate Structure Constraint, do not fit in the IBSP class. I conclude that such constraints are separate from island constraints and merit a distinct treatment, in line with the recent literature that analyzes them from a prosodic point of view.

1.3 Structure of This Dissertation

The structure of this thesis is as follows:

Chapter 2 of the thesis focuses on the formal preliminaries laying the ground for the chapters to come. It discusses the formal framework of Minimalist grammars (Stabler, 1997, 2011), adopted for syntactic analysis. Moreover, it delves deeper into the world of subregular literature explaining in detail the classes relevant for the phenomena covered in the thesis. The focus of this chapter is more on intuitions and provides numerous examples from natural languages for each class that is discussed. The formal definitions are provided in the appendix A for the math-oriented readers.

Building on the works of Graf and Shafiei (2019) and Shafiei and Graf (2020), chapters 3 and 4 discuss different syntactic dependencies, comparing various data structures.

More specifically, chapter 3 covers the first type of dependencies mentioned in the previous section, namely the *licensing* type dependencies. This chapter introduces a new mechanism that converts the key tree relation used for such dependencies, i.e. c-command, to a string-based representation. The investigation of generative capacity of such phenomena is done over *c-strings* then.

Chapter 4 focuses on another controversial topic of syntactic islands. Islands can be thought of as being dependencies with blocking effects, and therefore, require a different treatment. This chapter shows that, although still subregular over strings, islands belong to another subclass compared to c-command licensing dependencies.

Chapter 5 concludes the thesis, touches upon some advantages and limitations of string-based representation, and discusses directions for future work.

Chapter 2

Formal Preliminaries

Formal work on languages requires utilizing some formal frameworks. In this chapter, I lay the foundation for the discussions to come by providing some background information about the formal frameworks used throughout the thesis. The discussion includes Minimalist grammars (MGs; Stabler, 1997, 2011), as a formal theory of syntax. I further explain the Formal Language Theory as "... a measuring stick for linguistic theories (Jäger and Rogers, 2012: 1956)...", and its implications in terms of classifying language phenomena in a computationally hierarchical way. Relevant *subregular* language classes in the hierarchy are also explained in this chapter. Let us begin with MGs first.

2.1 Minimalist Grammars

In order to assess the complexity of syntactic dependencies, we need a formal model of syntax. I choose Minimalist grammars (MGs) to serve this purpose. MGs, inspired by Chomsky (1995), are a highly lexicalized, mildly context-sensitive formalism, for which structure building operations are assumed to be feature driven.

Based on Chomsky's Minimalist syntax (Chomsky, 1995), MGs are a derivational grammar formalism for building tree structures by combining feature-annotated lexical items via the operations *Merge* and *Move*[1].

Figure. 2.1 gives a concrete example of this process, comparing a more familiar phrase

[1]MGs implement only the core aspects of Chomsky (1995)'s Minimalist program. In terms of their faithfulness to the original framework, it is khown that they are fairely faithful, and the differences between the two are superficial. For a more detailed discussion on this matter, please refer to Section 2.3 of Graf (2013a).

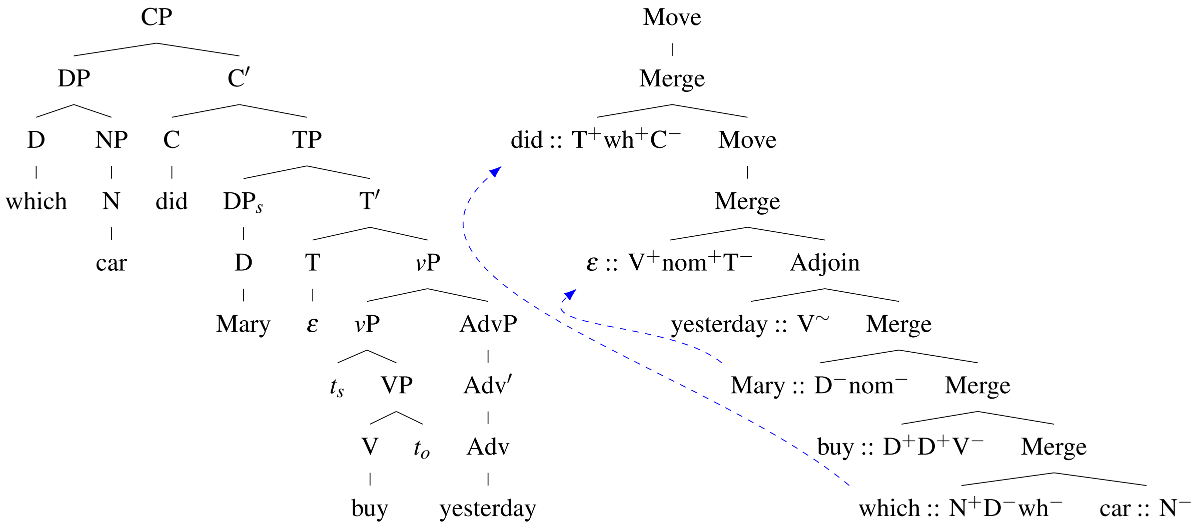


Figure 2.1: Phrase structure tree (left) and MG derivation tree (right) representations for *Which car did Mary buy yesterday*. The movement arrows in the derivation tree are included for reader's convenience, and are not part of the derivation.

structure tree with an MG derivation tree. Only a few key aspects of MGs matter for this thesis (see Stabler (2011) for a more extensive discussion). Each lexical item consists of a phonetic component and a finite string of features. There are four distinct types of features, two for the operation *Merge* and two for the operation *Move*. *Merge* and *Move* are the two main operations in MGs. The *Merge* features, shown with capital letters, consist of *category features* (X^-) and *selector features* (X^+), where X can be any category. On the other hand, the *Move* features, shown in lower case letters, include *licensee features* (f^-), and *licensor features* (f^+), where f can be any movement feature. To handle adjunction, following Frey and Gärtner (2002), I use an adjunction feature ($X^\sim$) which allows a lexical item to adjoin to an XP and give back an XP.

Notice that features differ in their case and polarity. As mentioned, *Merge* features are shown using upper case, whereas lower case is used for *Move*. Moreover, MGs assign each feature one of two polarities, *positive* and *negative*, and feature checking results in the deletion of two features of opposite polarity. The only feature that differs in this respect is the adjunction feature, which is shown by a tilde on the lexical item that acts as an adjunct, meaning that it selects a category of the type X and gives back the same category.

To give a concrete example, in our tree above, the verb *buy* takes an object and a subject and itself is a verb. Therefore, its feature system looks like *buy* :: $\mathrm{D}^+\mathrm{D}^+\mathrm{V}^-$, which encodes that the verb *buy* first selects a DP (the direct object), and then another DP (the subject). Only after *buy* has selected all its arguments can its category feature be targeted for selection. The 1-to-1 matching of selector and category features via *Merge* is called feature checking, and every lexical item must have all its selector and category features checked for the derivation to be legit (the head of the matrix CP is the only exception, as it is not selected by anything else).

The other two feature types drive the operation *Move*. A licensee feature, f^- indicates that the phrase headed by the lexical item undergoes f-movement, and the matching licensor feature, f^+, indicates the landing site of movement. As in Minimalist syntax, movement is a mechanism for displacing subtrees of an already assembled tree. Hence the head of a subject DP may take the form *the* :: $\mathrm{N}^+\mathrm{D}^-\mathrm{nom}^-$, and the subject landing site is provided by ε :: $\mathrm{V}^+\mathrm{nom}^+\mathrm{T}^-$, an unpronounced head of category T. A subject *wh*-mover would have two licensee features, e.g. *who* :: $\mathrm{D}^-\mathrm{nom}^-\mathrm{wh}^-$. Here *who* would first undergo subject movement, then *wh*-movement. A mover always targets the closest c-commanding head with a matching f^+. If two movers are forced to target the same f^+, the derivation is ill-formed. As with category and selector features, all licensor and licensee features must have been checked by the end of the derivation. Crucially, the mover stays in-situ in the MG system and only its feature percolates to higher nodes to be checked against the licensor feature. Another major difference is that the *wh*-mover (or any other mover) does not stop at any intermediate position. In other words, there is no successive cyclic movement in MGs. These differences turn out to be very important for me in this thesis, especially for explaining island constraints in Chapter 4. They will make it easier to define the dependency, or the blockers that block this dependency between the mover and its landing site.

One more feature that is relevant for me in this thesis is the adjunction feature. Instead of a category feature, a lexical item l may carry an adjunction feature $\mathrm{X}^\sim$, which allows it to adjoin to an XP. This checks the adjunction feature of l, but not the matching category feature on the head of the XP. In our example here, we have an adjunct *yesterday* with the adjunction feature $\mathrm{V}^\sim$, meaning that it adjoins to a VP and gives back a VP.[2]

Throughout the thesis, I represent derivation trees with the more compact format of dependency trees. The two are easily convertible to each other, however, dependency trees give us an easier way to obtain the dependencies we are interested in. To convert a deriva-

[2] I am ignoring the *v*P analysis for simplicity. It is not going to change the proposed analysis.

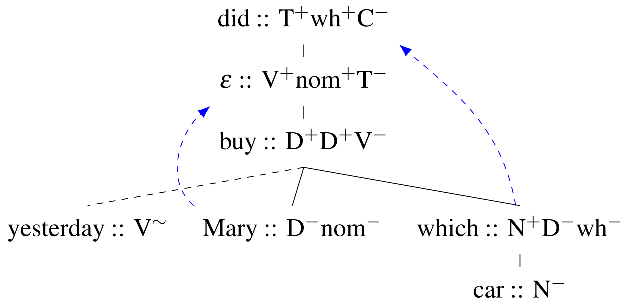


Figure 2.2: Dependency tree representation of MG derivation for *which car did Mary buy yesterday*

tion tree to a dependency tree, one needs to look at the thematic and head/argument relations between the lexical items. The arguments are dependents of the head that selects them. For instance, the verb *buy* selects for two NPs (or DPs) to be its object and subject. Therefore, these DPs are its dependents. To show this, these two DPs should be drawn as daughters to the verb. In other words, if a category feature (X^-) is selected by a selector feature (X^+), the former is a dependent of the latter. Having this in mind, the dependency tree for Fig. 2.1 is shown in Fig. 2.2. Note that arrows that indicate movement are included for reader's convenience, but they are not part of the derivation.

It is worth noting that using the features in the way explained here, we can obtain the dependency trees without having to look at or convert derivation trees. These trees basically encode the *selector-selectee* relationship in the tree. To obtain these trees, one has to look at which LI selects for which other LI. The former then becomes the mother of the latter in the tree. For instance, the verb *buy* :: $D^+D^+V^-$ selects for two DPs to be its object and subject. Therefore, these DPs are its dependents, and are daughters of the verb in the tree. In other words, if a category feature (X^-) is selected by a licensor feature (X^+), there is a daughter-mother relation between the two in the dependency tree. This will give us the same *dependency tree* for the sentence *which car did Mary buy yesterday* in Fig. 2.2 above. Recall that I use $X^\sim$ to show adjunction.

2.2 Obtaining C-command Relations as Strings (c-strings)

We are now familiar with the MGs and how they utilizes feature-annotated lexical items to obtain the *derivation trees* resembling the Minimalist Program's (Chomsky, 1995) *phrase* *structure* trees. I would like to focus on the central idea used in this thesis to determine the complexity of syntactic long-distance dependencies, which is representing these dependencies in a string format. Recall from the introduction in Chapter 1 that I am going to use two string representations to analyze binding and islands. In this section, I focus on one of them, namely *c-strings* and I leave the explanation for *a-strings* to §4.4.1.

I show how I will use an intermediate representation to handle c-command based computations. This representation, which I call *c-strings* is a string-based representation of c-command relations, which is attained via *dependency trees*, which themselves are easily obtainable from *derivation trees* by a regularity-preserving tree transduction[3]. These trees make it very easy to enforce c-command requirements for any given node, if we do not consider movement, which is discussed in §3.5. *Dependency trees* are also directly obtainable from the features on the LIs, without the need to do a conversion on *derivation trees*, as explained in the previous section.

2.2.1 Definition of C-strings

Encoding c-command from *dependency trees* is a simple task defined in Graf and Shafiei (2019) in the following way:

Definition 1. If u is a left sibling of v in an MG dependency tree t, then (the phrase headed by) u c-commands (the phrase headed by) v. In addition, v is c-commanded by every node that dominates it. (Graf and Shafiei, 2019: 207)

Let us apply this definition to our tree in Fig. 2.2 above. If we consider the node *Mary* to be the u in the definition, *Mary* c-commands its right sibling *which*, the v in the definition. Our v node, *which*, is also c-commanded by every node that dominates it, i.e. *buy*, ε, and *did*. In other words, *which* is c-commanded by its left sibling and its mother nodes. This essentially amounts to going left and up the tree to extract the c-commanders of a given node.

We can keep track of the c-commanders of a node in a string-based format. For instance, for our sample node *which*, its list of c-commanders is: *Mary buy* ε *did*. This process can be used for every node in the *dependency tree*, so that we can have a list of nodes that are

[3]For this reason, the *dependency trees* are also regular tree languages, just like their counterpart *derivation trees* (Michaelis, 2001; Kobele et al., 2007).

related to each other via c-command. This way of encoding and storing the c-command relation is formally defined by Graf and Shafiei (2019) below, as *c-strings*.

Definition 2. Let T be a tree such that node m has the daughters $d_1, \ldots, d_i, d, d_{i+1}, \ldots, d_n$, with $n \geq 0$. The *immediate c[ommand]-string* $ics(d)$ of d is the string $d\ d_i \cdots d_1$. For every node n of T, its *c[ommand]-string* $cs(n)$ is recursively defined as shown below, where $\cdot$ indicates string concatenation:

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

(Graf and Shafiei, 2019: 207)

The *command-strings* or in short *c-strings*[4] keep track of the c-command relations between different nodes in a *dependency tree* in a string-based format. Looking at our tree in Fig. 2.2, we have the following list of *c-strings* for different nodes:

- ics(car)= car :: N$^-$
- cs(car)= car :: N$^-$ which :: N$^+$D$^-$wh$^-$ Mary :: D$^-$nom$^-$ buy :: D$^+$D$^+$V$^-$ ε :: V$^+$nom$^+$T$^-$ did :: T$^+$wh$^+$C$^-$
- ics(which)= which :: N$^+$D$^-$wh$^-$
- cs(which)= which :: N$^+$D$^-$wh$^-$ Mary :: D$^-$nom$^-$ buy :: D$^+$D$^+$V$^-$ ε :: V$^+$nom$^+$T$^-$ did :: T$^+$wh$^+$C$^-$
- ics(Mary)= Mary :: D$^-$
- cs(Mary)= Mary :: D$^-$nom$^-$ buy :: D$^+$D$^+$V$^-$ ε :: V$^+$nom$^+$T$^-$ did :: T$^+$wh$^+$C$^-$
- ics(buy)= buy :: D$^+$D$^+$V$^-$
- cs(buy)= buy :: D$^+$D$^+$V$^-$ ε :: V$^+$nom$^+$T$^-$ did :: T$^+$wh$^+$C$^-$
- $ics(\varepsilon)$= ε :: V$^+$nom$^+$T$^-$
- $cs(\varepsilon)$= ε :: V$^+$nom$^+$T$^-$ did :: T$^+$wh$^+$C$^-$
- ics(did)= cs(did)= did :: T$^+$wh$^+$C$^-$

[4]Brody (2015) defines syntax in string format using dominance in the form of p(recedence)-strings and c(onceptual)-strings, which should not to be confused with c-strings in this thesis that encode c-command.

One can verify that these definitions of c-command extraction actually give us more than the list of c-commanders. These definitions predict a symmetrical and mutual c-command between heads and their specifiers. However, in the traditional sense, while specifiers c-command their heads, the opposite is not necessarily true. This notion of c-command therefore is a hybrid of c-command and m-command (Aoun and Sportiche, 1983), and is called *d[erivational]-command* by Graf and Shafiei (2019). We are going to see throughout this thesis that this difference is immaterial to the analysis and therefore, trivial.

The idea behind using such intermediate representation is to have a newer look at syntax and syntactic dependencies through a different lens. Brody (2015) is another work in stringifying syntax. Arguing that dominance is too powerful as a base syntactic relation, he proposes that one can consider syntax to use a more impoverished linear (string) orders rather than dominance trees. Such representation would mean that a sentence will have to correspond to not one but a collection of syntactic structures, similar to the *c-strings* proposed in the current thesis.

He uses precendence as the base relation, which he defines on ninety-degree rotated syntactic trees called "p(recedence)-trees", or *p-trees*. The "p(recedence)-strings", or *p-strings*, are then sets of nodes that precede a terminal node T in a p-tree. He defines licensed *p-strings* as the ones that express the concept of an extended word, which he names "c(onceptual)-strings". For a *p-string* to be well-formed, all its *c-strings* should be licensed as well. In Brody (2018), he elaborates more on the precedence specifiying that precedence in spellout is precendce in syntax. In order to ensure that sister nodes cannot interrupt each other and their spellout order respects constituency relations, Brody (2018) requires that if A immediately precedes B in syntax then X can intervene between A and B in spellout only if A precedes X (also) in syntax. Movement happens to a preceding node on this *p-tree* and this corresponds to c-command.

A valid *p-string* is then the one that begins with the initial symbol S (presumably the root node) and ends with a terminal symbol T. Moreover, *p-strings* themselves consist of sub-parts, which express the concept of extended word. These sub-parts, which he refers to as "c(onceptual)-strings", or is short *c-strings*, should be well-formed for a *p-string* to be valid too. Therefore, the validity of a *p-string* can be assessed recursively by accepting strings that are the results of the concatenation of a segment of a *p-string* with a valid *c-string*. Brody (2015) argues that the result of such concatenation is also a *p-string*.

To elaborate, given that *p-strings* start with S and end with T, the first segment of a given *p-string* that a *c-string* can concatenate with is the initial segment of that *p-string*. Moreover, for any *p-string* to be well-formed, it should end with a T symbol, and so every *c-string* should end in a T symbol or a terminal node.

In Brody's system, c-command becomes natural and central because i) the *p-strings* are paths defined from the initial symbol to the terminal nodes that do not take constituency into account, and ii) there is a monotony requirement that fillers must precede gaps in a movement operation dictating the order of sister nodes in *p-strings*.

Both *p-strings* defined in Brody (2015) and *c-strings* defined here encode c-command. However, the first difference between the two is this notion of asymmetric c-command. As previously mentioned, the *c-strings* I defined in this chapter encode a symmetric c-command relation between a head and its specifier, and are more in line with Aoun and Sportiche (1983) m-command.

Moreover, Brody (2015) argues for the centrality of c-command and against the redundancies of syntactic trees. It attempts to address the missed generalization that movement links c-commanding or dominating positions, which is a result of the requirement that movement has to be leftwards and upwards. On the other hand, my motivation is to argue for unifying accounts for various syntactic dependencies and to also show the cognitive parallelism between different linguistics modules. The end representations are similar in that in both works, syntax (or syntactic relations) is represented as a set of well-formed strings. These strings encode c-command for me, and they encode precedence with an emerged c-command for Brody (2015). His approach shares similar insights to mine in that syntax can be one-dimensional. Nonetheless, my approach here is motivated by the lack of unifying accounts and the complexity of various relations.

The goal is to not only show that syntax is much less complex than believed and shows parallelism with phonology and morphology, but more importantly, that the complex phenomena can be unified using a more formal analysis. To this end, in the next section, I discuss the *subregular* program and explain some of its implications in the modern day linguistics. I provide some natural language data and show how they fit in different classes of the *subregular* program based on their complexity.

Following that, I show how this new string-based representation of the c-command relations would allow us to use the computational machinery used for strings in phonology and morphology to account for c-command dependencies in syntax. This idea is used in Chapter 3 to analyze binding conditions and in Chapter 4 to analyze islands using a slightly different string-based representation.

2.3 Subregular Hierarchy

Formal language theory provides a measuring stick for defining languages mathematically and determining the computational complexity by using finite grammars. This field was initiated by Chomsky (1956) that characterizes, what is known today as, the Chomsky Hierarchy. This hierarchy classifies language patterns into four levels of hierarchy in terms of their complexity, namely *regular*, *context-free*, *context-sensitive* and *computably enumerable* languages. This hierarchy has gone through refinements in the past decades. The class of *subregular* languages occupies the space between the class of *finite* languages that can express only a finite number of strings, and the class of *regular* languages, which is believed to be an upper bound for phonological processes (Kaplan and Kay, 1994). Fig. 2.3 shows a more refined version of the original Chomsky Hierarchy.

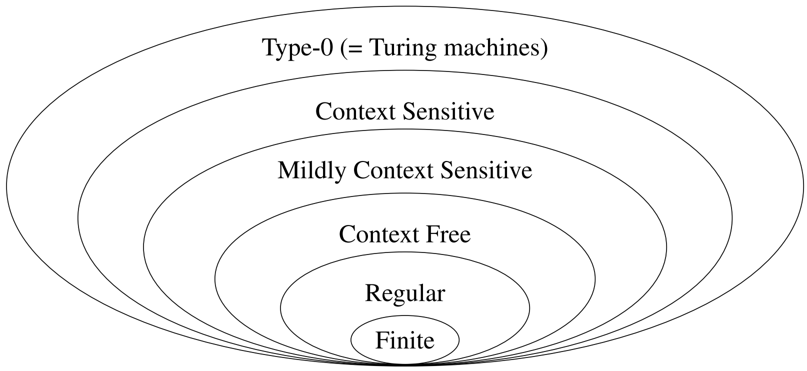
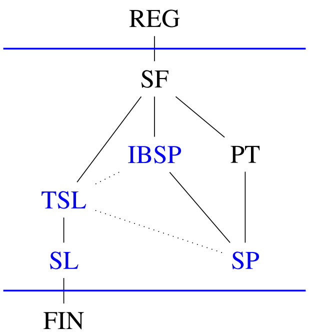


Figure 2.3: Chomsky Hierarchy (adapted from Jäger and Rogers (2012))

The class of *subregular* languages finds its importance in the recent works in the field of formal language theory, which have shown that many language patterns are less complex and are in fact subsumed within the regular class. This has been shown for phonology (Heinz, 2018), morphology (Aksënova et al., 2016; Chandlee, 2017), as well as syntax (Graf, 2018b; Graf and Shafiei, 2019; Vu et al., 2019), among others. The *subregular* class is divided into sub-classes on its own, which can be seen in Fig. 2.4. This figure shows how the region between the regular class of language and the finite languages can be divided into subclasses based on their properties and complexity. Parent classes subsume their children and are more powerful than them, while classes that are in the sibling relation are not known to subsume each other. More sub-classes get added to the *subregular* hierarchy as more linguistic phenomena are studied formally. Fig. 2.4 represents more prominent ones at the time of writing this thesis, and the ones that are more relevant to the theme of the current work are marked in blue.



Abbreviations

REG	Regular
SF	Star Free
IBSP	Interval-Based Strictly Piecewise
TSL	Tier-based Strictly Local
PT	Piecewise Testable
SL	Strictly Local
SP	Strictly Piecewise
FIN	Finite

Figure 2.4: Subregular Hierarchy (adapted from Aksënova, 2020)

The three more prominent *subregular* classes attested in phonology are *SL*, *SP*, and *TSL*. Local segmental patterns in phonology are SL (Heinz, 2010), whereas long-distance patterns are either SP or TSL based on the existence of a blocker or lack thereof (Heinz et al., 2011). TSL has been proposed to encompass "... all humanly possible segmental phonotactic patterns (Heinz et al., 2011: 62)". More recently, the class of *IBSP* was proposed to account for long-distance phonological dependencies that need to occur within specific domains (Graf, 2017, 2018a). I have a rather intuitive approach in the coming sections, whereby I provide examples from phonology and/or syntax to elaborate on different *subregular* classes. The formal definitions are provided in the Appendix §A, for the curious reader to refer to. The information in the appendix is not required to follow the discussions throghout the thesis.

2.3.1 Strictly Local (SL)

As the name suggests, the class of SL languages account for local dependencies across variety of language phenomena. To do so, there needs to be a defined SL grammar which either bans or allows certain substrings of a given length in the given language. In other words, the strings are assessed on a basis of *n*-grams defined in the SL grammar. If the grammar is positive, it allows substrings of the length n. However, if it is negative, it bans the substrings of length n.

Having this grammar, one can evaluate whether or not a given string belongs to the language by looking at its substrings of a pre-defined length and comparing them to the pre-defined grammar. If all the substrings of a given length belong to the positive grammar or are non-existent in the negative grammar, that string is well-formed; otherwise, it is illicit.

For example, a negative grammar such as G^{-} ={ab} shows that if a substring of length 2 that matches *ab* in a given string, that string is ill-formed. As a result, while *baaaa*, *aaaa*, and *bbbbbaaa* are well-formed, *aaaaab* or *abaaaa* are not. The reason for that is none of the well-formed strings have the bi-gram *ab* in them, while the ill-formed ones do, as in *aaaa**ab*** or ***ab**aaaa*. Since the set of forbidden factors is finite, using negative grammars is more often easier.

Some examples of the phenomena that fit into the SL class are discussed below. For a formal definition of this class, refer to §A.1.

Example 1. Final-obstruent devoicing: Final-obstruent devoicing, first formalized for German in Vennemann (1968), is observed in many languages such as Catalan, Russian, Polish (see Dinnsen (1985) and references therein), to name some. This phenomenon results in the word-final obstruents becoming voiceless. For example, while [ba:t] 'bath' is well-formed in German, [ba:d] is not. This phenomenon is SL_2 since we can use bi-grams (substrings of length 2) to evaluate the well-formedness of a word, and rule out the ones that end in voiced segments followed by word-edge marker, shown by $\ltimes$ symbol for the right edge and $\rtimes$ for the left edge. What this means is that we can forbid 2-factors such as {d$\ltimes$, v$\ltimes$, z$\ltimes$, ...}. Now, looking at our example, and augmenting our two words with word-edge markers, and applying the rule, we can correctly rule out the ill-formed [ba:d], while accepting the well-formed [ba:t].

$$^{*}\rtimes\ \text{b}\ \text{a:}\ \boxed{\text{d}\ \ltimes} \qquad\qquad ^{ok}\rtimes\ \text{b}\ \text{a:}\ \boxed{\text{t}\ \ltimes}$$

Figure 2.5: Final-obstruent devoicing is SL-2 with forbidden factor of "[+voice]$\ltimes$"

Example 2. Intervocalic lenition: Sound changes, especially occurring intervocalically, are referred to as lenition. Some examples of intervocalic lenition are stops targeted for spirantization, attested in Tiberian Hebrew and Spanish, respectively. Another example is intervocalic voicing of voiceless stops, attested in Warndarang (Kaplan, 2010). Let us take the spirantization in Spanish, by means of which intervocalic stops are realized as fricatives. In other words, *b*, *d*, and *g* are realized as β, ð, and γ, respectively. So, the [d] sound in [dama] 'lady' will be realized as ð in [laðama] 'the lady'. This can be captured by an SL_3 grammar forbidding [d] and the other two sounds between two vowels, {ada, aba, aga, ...}, and evaluating tri-grams (substrings of length 3) in the words. This is shown in Fig. 2.6.

* l a $\boxed{\text{d a m}}$ a   *ok* l a $\boxed{\text{ð a m}}$ a

Figure 2.6: Intervocalic lenition is SL-3 with forbidden factor of "V[-voice, +stop]V"

Example 3. Sibilant harmony is not SL: Languages such as Inesetño Chumash show long distance sibilant harmony, whereby the sibilants in a given word agree in their anteriority feature with the rightmost sibilant in the word (Applegate, 1972). For example, the /s/ sound in the word /**s**+xalam+**ʃ**/ 'it is wrapped' is realized as [ʃ], as in [**ʃ**+xalam+**ʃ**] in an agreement with the [-anterior] feature of the last sibilant in the word. In other words, the co-occurrence of [s] and [ʃ] sounds is ill-formed in /**s**+xalam+**ʃ**/. While this particular example can in principle be captured by an SL_7 grammar, banning the 7-gram of *sxalamʃ*, this is not generalizable, as the dependency between these two relevant segments is unbounded, and there is no fixed size as to how many elements can appear between the two. For instance, this grammar cannot rule out the ill-formed /ha**s**xintilawa**ʃ**/ 'his former Indian name', as the dependency size between the two sibilants in this example is 11, rather than 7. Regardless of how we move our 7-gram window, it is not going to capture the two sibilants in one frame. We can expand the window to include this particular example, but this is not generalizable.

* $\boxed{\text{s x a l a m ʃ}}$ *ok* $\boxed{\text{ʃ x a l a m ʃ}}$

? h a $\boxed{\text{s i n t i l a}}$ w a ʃ   ? h a s i n $\boxed{\text{t i l a w a ʃ}}$

Figure 2.7: Sibilant harmony is not SL due to the unboundedness of the dependency

As it turns, due to the unboundedness of long-distance dependencies, such as the current example, SL falls short of accounting for them. Such dependencies are either accounted for by Strictly Piecewise (SP) or Tier-based Strictly Local (TSL) grammars, discussed later in this chapter.

The examples provided show that local dependencies in phonology are SL on strings. Similar processes show that syntactic local dependencies are also SL, but on trees. Take the operation *Merge* as an example.

Example 4. Merge is SL: As explained in §2.1, *Merge* is a feature-driven operation, by means of which a lexical item with a *selector feature* X^+ selects for another lexical item with the same category feature, X^-. For example, in the phrase *which car*, *which* has a N^+ feature, selects *car* with the N^- feature, and the operation *Merge* is solidified, as shown below. Graf (2012a) shows that by lifting constraints from string n-grams to tree n-grams, we can get SL constraints over sub-trees. Consequently, *Merge*, is SL_2 over sub-trees, with the forbidden factor shown below in Fig. 2.8, which bans the *Merge* of an LI with X^+ feature with another LI that lacks the corresponding category feature X^-. Note that this holds true for *Merge* without adjunction. *Merge* with adjunction is TSL (Graf, 2018b).

Figure 2.8: *Merge* is SL-2 over MG Derivation Trees

This section showed that local dependencies are SL. For phonological dependencies, we used strings as the form of representation, while for syntactic dependencies, we looked at trees. We also saw that non-local dependencies, such as sibilant harmony, are not SL. They are rather either TSL or SP. The following sections explain more in detail what phenomena these *subregular* classes can capture. I start with SP first and move to TSL next.

2.3.2 Strictly Piecewise (SP)

The SP class captures non-local dependencies without a blocking effect, by looking at subsequent segments. What this means is that instead of looking at substrings (or adjacent symbols) of a given string, just like what we did for SL, we look at various subsequences in a given string for the class SP. For example in a given string of *abcd*, there are 6 subsequences, namely {ab, ac, ad, bc, bd, cd}. Similar to the SL class, we define either a positive or a negative grammar and evaluate the well-formedness of a given string by assessing its subsequences of a given length against our grammar. If none of these subsequences belong to the negative grammar, the string is well-formed. For a formal definition, please refer to §A.2.

We saw in the previous section that sibilant harmony cannot be captured by an SL grammar. In the next example, I show that this phenomenon is in fact SP, with the factor of 2, meaning that the subsequences of length 2 should be considered and evaluated for this phenomenon.

Example 5. Sibilant harmony is SP: We saw in Example 3 that this phonotactic long distance dependency is not SL. In this operation, the relevant segments are all the sibilants in a given word/string. In other words, all other segments appearing between sibilants are irrelevant. Therefore, one can only take sibilant subsequences into account and ignore everything else. While *s...s* or *ʃ...ʃ* are well-formed subsequences or 2-factors, *ʃ...s* or *s...ʃ* are not. In other words, the constraints **ʃ...s* and **s...ʃ* appear to be in effect. This makes sibilant harmony, without blocking, an SP_2 grammar.

* s x a l a m ʃ $\qquad\qquad$ ^{ok} ʃ x a l a m ʃ

Figure 2.9: Sibilant harmony without blocking is SP with forbidden factors such as *s...ʃ*

Example 6. Unbounded Tone Plateauing (UTP) is SP_3: Another phenomenon that can be captured by an SP grammar is UTP. This operation prohibits a low tone (L) to appear between two high tones (H). What this means is that the subsequence of H-L-H is prohibited, in languages such as Kihunde and Mamaindé (see Hyman (2011) and references therein). Therefore, while HHH or LHL or HHHL are well-formed, HLH or HLLLH or LHLH are not. As mentioned, it is the **HLH** sequence that is forbidden in such languages regardless of how many Ls appear in between the two Hs. This is captured by an SP_3 grammar.

Example 7. Sibilant harmony with blocking is not SP: The Berber language family shows long distance sibilant and voicing harmony, with and without blocking effects. An

* H L L L H $\qquad\qquad$ ^{ok} H H H L

Figure 2.10: UTP SP with a forbidden factor of HLH

example of assimilation without a blocking effect is Tamajaq Tuareg of Niger, which exhibits a voicing alteration of sibilants paired with anteriority assimilation. This results in the complete identity between the sibilant in the prefix and the sibilant in the root that triggers this assimilation (Hansson, 2010). Some examples are provided in (1). Similar to harmony in Inesetño Chumash, this can be captured by an SP grammar that forbids subsequences of *s...z* and *z...s*, etc, when the two elements do not show voicing and anteriority harmony. Note that the underlying form of the prefix is [s], as evident by examples in (1a).

(1) Tamajaq Tuareg of Niger: harmony without blocking

a.	əlməd	‘learn, study’	s-əlməd	‘teach, inform’
	əwəs^{ʕ}	‘boil (intr.)’	s-əwəs^{ʕ}	‘boil (tr.)’
	qusət	‘inherit’	s-əqːusət	‘cause to inherit’
b.	əntəz	‘pull out, extract’	z-əntəz	‘cause to extract’
	guləz	‘be left, remain’	z-əgːuləz	‘cause to remain’
c.	măʃăn	‘be overwhelmed’	ʃ-əmːəʃən	‘overwhelm’
	fərəʃːt	‘be ugly, humiliated’	ʃ-əfərəʃːt	‘make ugly, humiliate’
d.	kuʒə	‘saw (v.)’	ʒ-əkːuʒət	‘cause to saw’
	ăʁʒu	‘be amazed’	ʒăʁʒu	‘amaze’

(Hansson, 2010, ex.6)

However, other languages, such as Imdlawn Tashlhiyt, show a similar harmony but with blocking effects, (2). In these languages, the harmony process is blocked if a voiceless obstruent intervenes between the two sibilants (reported in Hansson, 2010, based on data from Elmedlaoui, 1995). In (2), the examples in (2a) and (2b) show that voiceless obstruents block the voice feature to propagate across them. This blocking effect only affects the voicing assimilation, the assimilation of anteriority feature is unaffected by these elements. Voiced obstruents are transparent as shown in the examples in (2c). This blocking effect cannot be captured by an SP grammar. The SP_2 grammar above prohibits *s...z* and *z...s*. This grammar would incorrectly rule out well-formed words such as [sħuz] since the subsequence *s...z* is observed in this word.

(2) Imdlawn Tashlhiyt: voicing harmony with blocking

a.	ħuz	s-**ħ**uz	'annex'
	ukz	sː-u**k**z	'recognize'
b.	fʒːiʒ	ʃ-**f**ʒːiʒ	'to go for a walk'
	qːuʒːi	ʃ-**q**ːuʒːi	'be dislocated, broken'
c.	bruzːa	z-**b**ruzːa	'crumble'
	$\text{g}^{\text{wʕ}}\text{r}^{\text{ʕ}}\text{a}^{\text{ʕ}}\text{z}^{\text{ʕ}}$	$\text{z}^{\text{ʕ}}$ː-$\mathbf{g}^{\text{wʕ}}\text{r}^{\text{ʕ}}\text{a}^{\text{ʕ}}\text{z}^{\text{ʕ}}$	'regret'

(Hansson, 2010, ex.11)

Example 8. Dissimilation with blocking is not SP: Latin has an unbounded dissimilation rule changes [l] to [r] in the suffix *-alis* if it is preceded by [l], examples in (3).

(3) Latin Liquid Dissimilation (without blocking)

a.	nav-alis	'naval'
b.	milit-aris	'military'
c.	lun-aris	'lunar'
d.	lupan-aris	'whorish'

(Jensen, 1974: 679)

This dissimilation is blocked if there is another [r] between the target and the trigger, as shown in (4).

(4) Latin Liquid Dissimilation (with blocking)

a.	flor-alis	'floral'	*flor-aris
b.	sepulkr-alis	'funeral'	*sepulkr-aris
c.	litor-alis	'of the shore'	*litor-aris

(Odden, 1994: 314)

Similar to what we saw for Example 7, this dissimilation cannot be accounted for using SP grammar. If we ban the subsequence *l...l*, we can correctly rule out the dissimilation cases without blocking, but this forbidden factor would also predict that words like *floralis* to also be ill-formed, which is not the case. We will see that this phenomenon as well as Imdlawn Tashlhiyt harmony are TSL.

We have seen so far that local dependencies, such as word-final devoicing and intervocalic lenition, fit the class of SL. Moreover, non-local dependencies without blocking effects, such as sibilant harmony and UTP, belong to the SP class. Non-local dependencies without blocking effects can be accounted for by another *subregular* class, namely TSL, which also captures non-local dependencies with blocking effects. This class is discussed in the following section. As usual, not knowing the formal definition is not detrimental in following the discussion in the upcoming section.

2.3.3 Tier-based Strictly Local (TSL)

Intuitively, one can say that the class of TSL is a blend of the class SL and the class SP. The TSL class is similar to SP in the sense that it looks at non-local dependencies. It is also similar to SL in the sense that it evaluates these non-local dependencies in a local fashion by putting the relevant segments on a tier and in the proximity of each other. In other words, TSL languages are string languages that are SL once one masks out all irrelevant symbols. And although this class assesses long-distance dependecies, similar to SP, it is distinct from it as it can also be used to model non-local dependencies with blocking effects, something that the SP class is incapable of accounting for. For a formal definition of TSL, please refer to §A.3.

Let us look at the sibilant harmony in Berber languages again and see how TSL can capture this long-distance dependency. Remember that some of the data, i.e. harmony with no blocking, falls under the SP class. The same data group is also TSL. However, we will see how the harmony with blocking effects is also TSL, something SP cannot capture.

Example 9. Sibilant harmony with no blocking is TSL: To capture a long-distance assimilation process using TSL, we need to forbid disagreeing factors on a tier (Heinz et al., 2011). Therefore, sibilant harmony can be accounted for by using forbidding factors such as *sz* or *sʒ* on the sibilant tier. Consider our example in (1) again, repeated below in (5) for convenience.

(5) Tamajaq Tuareg of Niger: harmony without blocking

a.	əlməd	‘learn, study’	s-əlməd	‘teach, inform’
	əwəsˤ	‘boil (intr.)’	s-əwəsˤ	‘boil (tr.)’
	qusət	‘inherit’	s-əqːusət	‘cause to inherit’
b.	əntəz	‘pull out, extract’	z-əntəz	‘cause to extract’
	guləz	‘be left, remain’	z-əgːuləz	‘cause to remain’
c.	măʃăn	‘be overwhelmed’	ʃ-əmːəʃən	‘overwhelm’
	fərəʃːt	‘be ugly, humiliated’	ʃ-əfərəʃːt	‘make ugly, humiliate’
d.	kuʒə	‘saw (v.)’	ʒ-əkːuʒət	‘cause to saw’
	ăʁʒu	‘be amazed’	ʒăʁʒu	‘amaze’

(Hansson, 2010, ex.6)

TSL can easily capture this harmony, as illustrated in Fig. 2.11 below. Since the sibilants are the elements in question, we only need to project them on the tier. The dotted lines show tiers. The symbols written on them specify the features or the elements with those features that need to be projected on the tier. The [ant, voice] symbols on the dotted lines correspond to anteriority and voicing features, respectively. While having two identical sibilants on a tier is legitimate, a ban on non-identical sibilants would rule out illegitimate words. In other words, a ban on *sz* or *sʒ* suffices to rule out the ill-formed forms of the prefix.

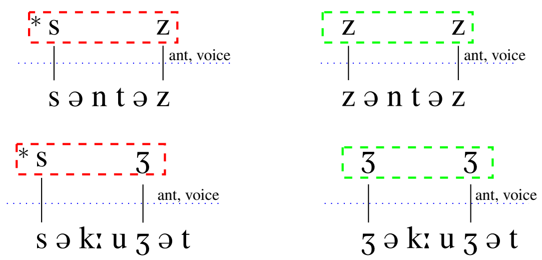


Figure 2.11: Long-distance sibilant and voicing harmony is TSL over strings as it can be expressed as a local ban against *sz* and/or *sʒ* on a voicing+anteriority tier.

As seen earlier, this harmony process can be captured by an SP_2 grammar that bans the *s...z* or *s...ʒ* subsequences. However, TSL is also able to capture long-distance dependencies with blocking effects, which SP is unable to account for.

Example 10. Dissimilation with blocking is TSL: As mentioned previously, Latin unbounded dissimilation rule changes [l] to [r] in the suffix *-alis* if it is preceded by [l]. This process is blocked if there is another [r] between the target and the trigger. This process is easily captured by a TSL grammar, which bans a *ll* sequence on the lateral tier. This would correctly rule out forms such as /lunalis/, because the /l/ sound in the suffix has not surfaced as [r] as expected, since two /l/ sounds are adjacent on the tier. Moreover, it correctly accepts forms such as [lunaris], with the expected dissimilation process and [floralis], which exhibits the blocking effect of [r]. The latter case is accepted due to the fact that the two [l] sounds are not adjacent on the lateral tier and the forbidden factor of *ll* is not instantiated.

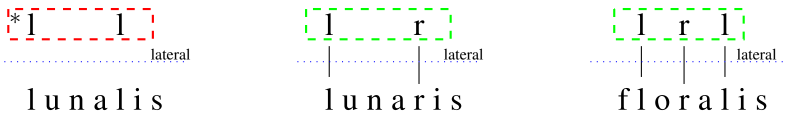


Figure 2.12: Long-distance Latin dissimilation with blocking is TSL over strings as it can be expressed as a local ban against *ll* on a tier of liquids.

Similar to what we did for SL, providing both phonological and syntactic examples, I would like to give an example of a syntactic phenomenon that is TSL over trees.

Example 11. Move is TSL: Graf and Heinz (2015) show that *Move* is TSL over MG trees. Graf (2018b) reaches the same conclusion by using a slightly different viewpoint, and the following two constraints on *Move* tier:

- Every *Move* node has exactly one LI among its daughters; i.e. every f^+ has exactly one f^- among its daughters.
- Every LI has a *Move* node as its mother; i.e. every f^- must have one f^+ mother.

We need to project all *Move* nodes with f^+ feature, as well as all the LIs with matching licensee feature, and nothing else.

Let us look at the derivation tree in Fig. 2.13 with two *nom* movers and one *wh* mover. To construct the *nom* tier, we project all the nom^+ together with all the nom^- nodes, and nothing else. To construct the *wh* tier, we project all the wh^+ and all the wh^- nodes, and nothing else. Since every nom^- has only one nom^+, and vice versa, the *nom* tier is well-formed. The same holds for the *wh* tier as well, with the wh^+ having only one wh^- daughter and the wh^- having only one wh^+ mother.

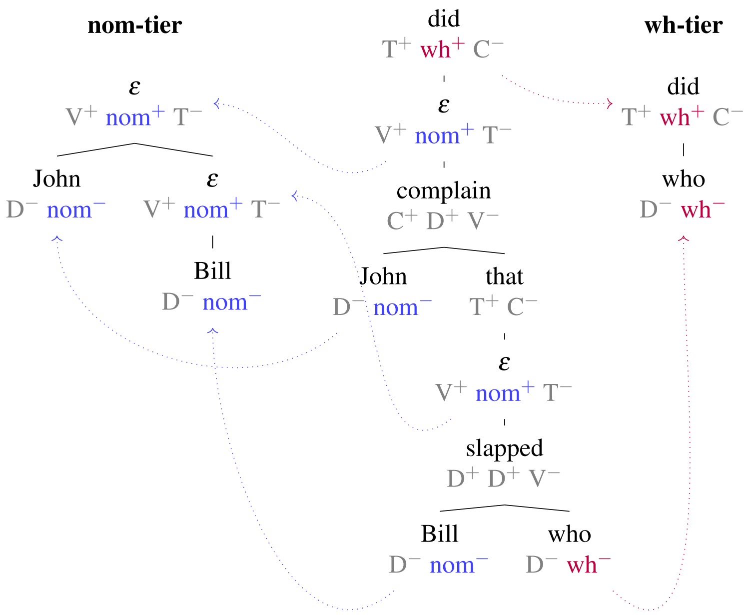


Figure 2.13: Well-formed *nom Move* tier (left) and and *wh Move* tier (right) over MG derivation tree

TSL can capture a variety of phonological long-distance dependencies. As a matter of fact, it has been proposed as a loose upper bound for computational complexity of such dependencies, where all dependencies are conjectured to belong to natural extensions of TSL (Heinz et al., 2011; Graf and Mayer, 2018). Although there are circumstances where TSL alone falls short of accounting for the data, all of these seem to fit into an extension of TSL. One example is sibilant and voicing harmony with blocking, which is in fact an MTSL (Multi-tier Strictly Local), rather than TSL. Among other phenomena that do not completely fall under TSL is Sanskrit n-Retroflexion, which is proposed to be IO-TSL (Input-Output TSL) (Graf and Mayer, 2018). Among these, *sibilant harmony with blocking* is discussed below, while the *Sanskrit n-Retroflexion* is discussed in the next chapter in §3.4.2.

Example 12. Sibilant harmony with blocking is MTSL: Previously, we saw that Imdlawn Tashlhiyt exhibits sibilant harmony, which is blocked by an intervening voiceless obstruent, as shows in (2) and repeated in (6). This process cannot be captured by SP, but it can be captured by TSL, more specifically, MTSL.

(6) Imdlawn Tashlhiyt: voicing harmony with blocking

a.	ħuz	s-**ħ**uz	'annex'
	ukz	sː-**uk**z	'recognize'
b.	fʒːiʒ	ʃ-**f**ʒːiʒ	'to go for a walk'
	qːuʒːi	ʃ-**q**ːuʒːi	'be dislocated, broken'
c.	bruzːa	z-**b**ruzːa	'crumble'
	$\text{g}^{\text{wʕ}}\text{r}^{\text{ʕ}}\text{a}^{\text{ʕ}}\text{z}^{\text{ʕ}}$	$\text{z}^{\text{ʕ}}\text{ː-}\mathbf{g}^{\text{wʕ}}\text{r}^{\text{ʕ}}\text{a}^{\text{ʕ}}\text{z}^{\text{ʕ}}$	'regret'

(Hansson, 2010, ex.11)

In order to capture blocking effects, we need to project potential blockers on the tier in addition to the sibilants. However, since having the obstruent blocker on the tier would break the tier locality required to model the anteriority assimilation, more than just a single tier is required to model this generalization. To do so, we need two different tiers, one with sibilants, and one with obstruents on a voicing tier. The sibilant tier assesses the anteriority harmony between the two sibilants, banning *sz* or *sʒ* sequences, just like what we saw for Tamajaq Tuareg's sibilant harmony without blocking. The tier for the voicing harmony forbids all combinations of voiced sibilants followed by voiceless obstruents (*zk, zf, zħ,* etc.).

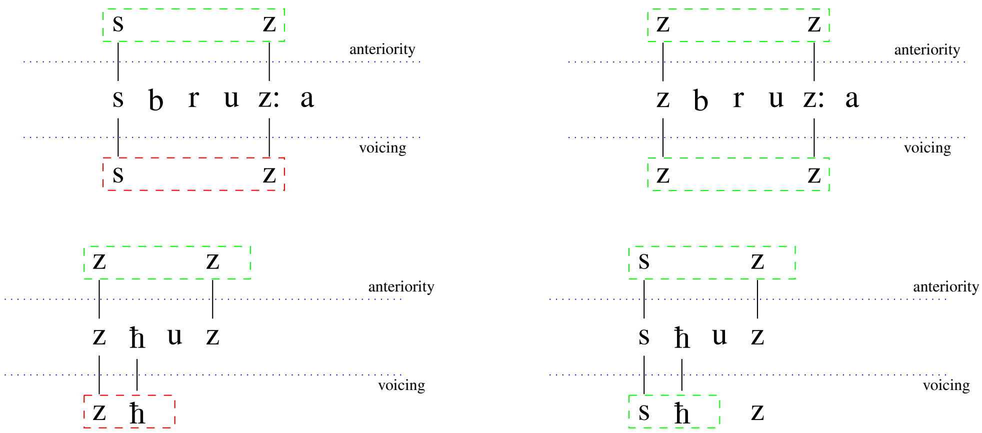


Figure 2.14: Long-distance sibilant harmony with blocking effect is MTSL over strings as it can be expressed as a local ban against *sz* on a tier of sibilants (upper tier), and a ban against *zħ* on a voicing tier (lower tier).

It is important to note that the ban on *sz* sequence, which ruled out illicit forms in the previous example, does not apply here. The tier projections for Imdlawn Tashlhiyt assess two different mechanisms separately, one is voicing and the other one is anteriority feature. This sequence, while ill-formed on a voicing tier, is well-formed on the anteriority tier. If there is no voiceless obstruent on the voicing tier, and these two elements are adjacent, then the word would be ruled out, e.g. /sbruzːa/. However, in an example like [sħuz], since [ħ] is projected on the voicing tier, these two sounds /s/ and /z/ are not adjacent to each other on that tier. Therefore, this word form is correctly accepted.

The previous sections lay the ground for what lies ahead in Chapter 3 and Chapter 4, where I use the foundations here to analyze binding relations and island constraints. We saw that the SL class captures local dependencies, while non-local dependencies are captured by either the SP or the TSL class. I also introduced an extension of the TSL class, namely MTSL, which accounts for harmony with blocking effect. Another extension of the TSL class is IO-TSL class, which is captured in §3.4.2. The IO-TSL class is the one that binding relations fit into. I will also introduce an extension of the SP class, i.e. IBSP in §4.3, to which the island constraints belong. In the next section, I explain why this is all relevant and how we can model *c-strings* in this sense. This will be relevant in Chapter 4, when I model *a-srtings* in a similar way.

2.4 Modeling c-strings in the Computational Sense

Let us see how *c-strings* can be modeled and evaluated in the computational format presented in the previous sections.

Let us start with SL and what phenomenon is SL over *c-strings*. In the section about SL §2.3.1, I showed that Merge is SL over derivation trees. I showed before that *c-strings* are obtained via dependency trees. Taking our dependency tree in Fig. 2.2, repeated here in Fig. 2.15, we can see that while the Merge relation in a derivation tree is a sibling relation, it amounts to a mother-daughter relation on a dependency tree. In a *c-string* format, Merge is equal to the elements on the right and the left of a given element. If the element in question has a *selector* feature of X^+, the element to its left should have the equivalent X^-. Moreover, if the element in question has a *selectee* feature of X^-, the element to its right would have the equivalent X^+. Depending on the number of the *selector* features, we need to move further to the left to see whether we find the equivalent X^- features on the elements. Let's take the examples below, with the *c-string* of *car* in our example.

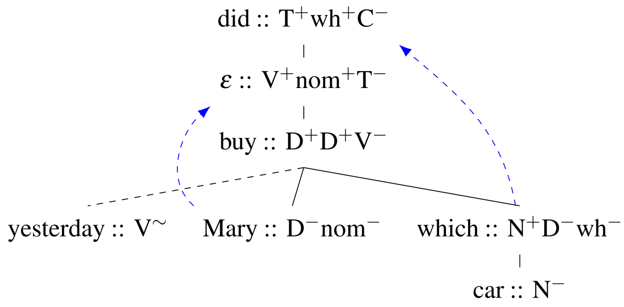


Figure 2.15: Dependency tree representation of MG derivation for *which car did Mary buy yesterday*

(7) *cs*(car)= car :: N^- which :: $\mathrm{N}^+\mathrm{D}^-\mathrm{wh}^-$ Mary :: $\mathrm{D}^-\mathrm{nom}^-$ buy :: $\mathrm{D}^+\mathrm{D}^+\mathrm{V}^-$ ε :: $\mathrm{V}^+\mathrm{nom}^+\mathrm{T}^-$ did :: $\mathrm{T}^+\mathrm{wh}^+\mathrm{C}^-$

For instance, let's evaluate the Merge operation fot the given element *Mary*. This element does not have a *selector* feature, so there is no need to look at the elements on its left. However, it has an D^- feature. Therefore, we need to make sure that it is merged with the right element that has the equivalent *selector* feature. Looking at the immediate element to its right, we can see that *buy* has D^+, which matches the D^- of *Mary*. In this sense, we can say that Merge is SL_2 over *c-strings*, whereby we should have the following configuration.

(8) **Merge over c-strings:** *a :: X^- b :: Z^+ (Not accurate)

However, we can immediately reject this, as this does not hold true for the element *which*, as the element immediately following it does not have the matching *selector* feature. Rather, it is the second element to the right that has the matching feature. So, maybe Merge is not SL over *c-srtings*. What if we look at the opposite direction and consider the elements to the left of a given element. Taking the element *buy* as an example, we can see that it has two D^+ feature, and looking at the two elements to its immediate left, we see that they both have the matching D^- feature. We can conclude that the Merge is allowed for these three elements. But, that would push our SL factor to 3 instead of 2. Besides, it does not say much about elements without any *selector* feature, such as *Mary*. We can probably say that for such elements, only consider what is on their immediate right, and for elements with *selector* feature(s), look at the same number of elements to their left as the number of their *selector* features.

This has become convoluted so quickly, though. While we can probably come up with rules to assess Merge with *c-strings* in the limit of SL class, it does not seem as straightforward as the SL grammar proposed over MG derviation trees for this operation. The goal here is not to come up with such a mechanism, rather it is to familiarize the reader with the steps taken to come up with such a mechanism for a given representation. It is also to show how *c-strings* will be and are evaluated throughout this thesis for a given phenomenon.

Let's see if we can model Move with *c-strings*. Again, taking our previous example into account, we have two movers, *which* :: $\mathrm{N}^{+}\mathrm{D}^{-}\mathrm{wh}^{-}$ and *Mary* :: $\mathrm{D}^{-}\mathrm{nom}^{-}$. Move is a long-distance dependency by definition. Therefore, we can directly rule out the use of SL for this operation. The two other classes that account for long-distance dependencies are SP and TSL. Let's see if we can model Move over *c-strings* in the SP class. We know by now that SP grammars look at subsequences on a given string and evaluate them against a given grammar. The movers in MGs have f^{-} features that should be matched with a matching f^{+} feature regardless of the distance between them. In that sense, Move (in its basic sence) is SP over *c-strings* since if we find a matching f^{+} feature for each f^{-}, the string is well-formed. For the movers in our example, we can see that we have the matching *licensor* features for each of them.

(9) **Move is SP over c-strings:** a :: f^{-} b :: f^{+}

It is important to know that there should only be one and only one such matching feature, otherwise, the string is ill-formed. The following constraint, the *Shortest Move Constraint* (SMC) indicates that we cannot have two elements with the same move feature in an MG derivation tree.

(10) **SMC:** Two licensee features may be both active at the same time in the derivation tree only if they are different. (Graf, 2013a: 25)

Having this in mind, the Move operation is pushed to be SP_3 over *c-strings* since instantiating one licit relation between a mover and its landing site suffices to rule the string as well-formed. The following configuration elaborates this point.

In this section, I showed how *c-strings* can be modeled and evaluated in the computational sense. These (and similar) intermediate representations are used to model syntactic dependencies such as binding and island constraints in the coming chapters, where I will show that unification of these conditions is possible using a formal viewpoint and a new look at how these dependencies are represented.

* a :: f^- c :: f^- b :: f^+ $\qquad$ ^{ok} a :: f^- c :: g^- b :: f^+

Figure 2.16: Move SP over *c-strings* with a forbidden factor of $f^-f^-f^+$

2.5 Conclusion

This chapter lays out the formal foundation required for the following chapters. It starts with the MGs as a formal framework for Minimalist syntax, showing that they consist of feature-driven operations, including two main operations of *Merge* and *Move* and a third operation of *Adjunction*. It then moves to touch upon some works on FLT, in particular the *subregular* framework.

Following that, I introduce the central idea behind this thesis, which is the use of string-based representation, as intermediate representations to assess various syntactic dependencies. This form of representation will allow us to provide a unified account of binding and syntactic islands from a formal point of view. In this chapter, I focused on *c-strings* as a new form to encode c-command relations. But, a similar idea is pursued for other dependencies such as islands in Chapter 4.

The *subregular* hierarchy, which has received a great deal of attention in the past decade, has made significant strides in the world of phonology. I discussed the bases of the *subregular* classes that are relevant to the phenomena in this dissertation. Therefore, the discussion on *subregular* classes started with SL, which captures local dependencies, and moved to TSL, which captures non-local dependencies by making them local on a tier. I also explained the class SP, which is used to capture unbounded dependencies without blocking effects. While this allows SP to readily capture such dependencies, having no notion of locality comes with its own shortcomings. In the following chapters, I show that these basic classes of SL, SP and TSL need to be expanded to IO-TSL and IBSP classes, to account naturally for binding relations and island constraints, respectively.

Chapter 3

Tier-based Locality of Binding

This chapter focuses on dependencies that rely on c-command to be licensed. The c-command relations between different nodes in a tree are obtained in the form of stringsets via a mechanism explained in (§2.2), and then the constraints are applied on these stringsets. Empirical coverage comes from binding conditions, and various proposals in this field.

The previous chapters set up the stage for the discussions in this chapter and the chapters to come. More specifically, what is needed for the current capture is to be familiar with MGs mechanism (§2.1) and also the *subregular* class of TSL (§2.3.3).

One of the most well-known syntactic phenomena that relies on c-command is binding principles. This section discusses binding principles, and various analyses of binding facts showing how they can be encoded using *c-strings*. Their *subregular* complexity is then examined in the following section. The focus is on principle A and reflexive binding. The discussion includes English as the base language and then moves on to include cross-linguistic variation.

Principle A of binding deals with *anaphors*. Anaphors are special type of referential noun phrases that need to pick out their references from other (linguistic) elements. The theoretical literature has attempted to define the environments and conditions for such noun phrases. However, the (apparently) diverse behavior of anaphors within and across languages has led to rather dissimilar analyses of these noun phrases. In this section, I start with a classical analysis of anaphors (§3.1), which dates back to Chomsky (1980). Local and long-distance bindings are discussed in this sense. While the classical binding analysis can readily explain short distance binding, it faces challenges when it comes to explaining long-distance binding. There are various approaches addressing these issues. I will discuss the LF movement approach by Pica (1987) (§3.2.2), and show that it can be restated in terms of *c-strings*. I will then move to discuss some more recent accounts that are more minimalist in nature than other, (§3.3). The interaction of movement and binding is discussed in §3.5. Each subsection ends with the modeling of the conditions using *c-strings*.

3.1 Classical Binding

In its very basic sense, principle A requires a reflexive pronoun to be bound in its binding domain[1]. What is meant by "bound" and "binding domain" has been a topic of scholarly discussions for the past decades.

Starting with the first one, "...an anaphor[2] α is bound in β if there is a category c-commanding it and coindexed with it in β; otherwise, α is free in β (Chomsky, 1980: 10)".

The second one, or the "binding domain" has proven to be a bit trickier to define. This difficulty stems from intra-language as well as cross-linguistic variation. The intra-language issue is that anaphors of the same shape tend to behave differently in different constructions, e.g. have local or long-distance antecedents. The cross-linguistic issue is also similar in that anaphors in different languages seem to require different binding domains. I start with English and local binding, and then move to non-local binding and cross-linguistic differences.

3.1.1 English Local Binding

In the simplest cases, such as the examples in (1), the binding domain seems to be limited to TP. The sentences are grammatical if the anaphor is able to find its antecedent within the TP that contains it.

(1) a. [$_{\text{TP}}$ Poirot$_i$ hurt himself$_i$].

b. * Poirot$_i$ thinks [$_{\text{CP}}$ that [$_{\text{TP}}$ Miss Marple hurt himself$_i$]].

[1]It is worth mentioning that the discussions in this chapter include only A-binding, or binding from A-positions in the sense of Haegeman (1994). Binding from A′-positions, like topicalized elements, are excluded.

[2]Anaphors are of two types: reflexives and reciprocals. The focus of this chapter is on reflexive anaphors.

c. * Poirot$_i$ believes [$_{CP}$ that [$_{TP}$ himself$_i$ is the best detective]].

(Haegeman, 1994: 208-216)

However, one can easily find examples that challenge this generalization. In Exceptional Case Marking (ECM) cases like (2), this generalization does not seem to hold, where the reflexive *himself* is bound by an element outside the TP containing it.

(2) Poirot$_i$ believes [$_{TP}$ himself$_i$ to be the best]. (Haegeman, 1994: 213)

It is reasonable to conclude that the difference lies in the tense of the TPs. While tensed clauses act as binding domains, the infinitival ones cannot. However, this prediction is not borne out. As the following example shows, the anaphor can be bound by an antecedent beyond the limits of the TP containing it.

(3) Poirot$_i$ believes [$_{CP}$ that [$_{TP}$ a picture of himself$_i$ will be on show at the exhibition]].

(Haegeman, 1994: 216)

What is more is that NPs (or DPs) can themselves act as binding domains as is shown in (4a). Although the anaphor and its antecedent are within the same TP, the binding relation between them is not actualized. It is concluded that DPs can act as binding domains as well. But as the contrast in (4) shows, it is not just any type of DP, rather the DPs that contain another DP in their specifier positions.

(4) a. * Poirot$_i$ believes [$_{DP}$ Miss Marple's description of himself$_i$].
b. Miss Marple believes [$_{DP}$ Poirot$_i$'s description of himself$_i$].
c. Poirot$_i$ believes [$_{DP}$ any description of himself$_i$]. (Haegeman, 1994: 213-214)

The traditional solution for such sentences attributed these to the notion of *accessible subjects*, proposing that phrases with subjects (overt or covert) act as binding domains. An overt subject is basically a DP in a specifier position. A covert subject, or big SUBJECT, corresponds to finite AGR, a feature on T (I) that warrants the agreement between the subject of the sentence and verb morphology. Incorporating this condition into the existing binding principle, we end up having (5) as our definition of *reflexive binding*.

(5) A reflexive X must be bound in the minimal domain containing X, X's governor[3] and an accessible subject/SUBJECT. (Haegeman, 1994: 222)

Having this definition in place, let us go through the seemingly problematic examples one by one. Starting with (2), I already mentioned that this sentence is an ECM case, with an infinitival embedded clause. The infinitival clause does not have an AGR feature and so the phrase headed by the embedded T is subject-less. Therefore, the domain of reflexive is the next clause, in this case the matrix TP. So, the matrix subject, can bind the reflexive in the embedded clause. To elaborate further, tenseless clauses might not have any form of a *subject*. The SUBJECT is mainly present in finite clauses to encode morphological agreement between the AGR head and the clausal subject. It is called the big SUBJECT to distinguish it from the NP *subject* of the clause. However, this is still not enough to account for grammaticality of (3).

This problem was originally explained using the *i-within-i* filter or circularity in reference in the sense of Chomsky (1981, 1993). This filter, provided in (6), bans coreference between a phrase and one of its proper subconstituents.

(6) * $[_{\gamma_i} \ldots \delta_i \ldots]$ (Chomsky, 1993: 212)

According to this filter, the phrase *a picture of himself* runs into the issue of circularity as shown below.

(7) $[_{\mathrm{NP_i}}$ a picture of $[_{\mathrm{NP_i}}$ himself$]]$

We already know that the subject of the lower clause is the whole NP phrase, hence the AGR head is related to the whole phrase. Since the configuration above is in violation of the *i-within-i* filter, the embedded AGR cannot be used to define the binding domain of the anaphor. In other words, there is no accessible subject in the lower clause to construct a binding domain, and so, the binding domain must be extended to the matrix clause. As a result, the matrix subject can bind the reflexive inside the embedded clause. The traditional binding theory and the *i-within-i* condition are not unproblematic. However, they can cover a vast variety of empirical data. However, the purpose of this thesis and this chapter is to show how computational modeling can shed light into the complexity of various syntactic

[3]Governors are the lexical heads (V, N, P, A) and tensed I (Haegeman, 1994: 193). This notion is trivial and negligible for the discussions in this chapter.

phenomena, and not to solve the edge cases in the binding theory; thus, I suffice with these findings. I use them as my base for modeling. Then I discuss an expansion to handle long-distance binding. I also discuss more recent works in the minimalist framework (§3.3). Before that, I show how we can model traditional binding theory for local anaphors using *c-strings*.

3.1.2 Modeling Principle A in Classical Sense

Let us begin with the simplest example of binding, i.e. a mono-clausal phrase with an object reflexive and a subject antecedent, such as (1a). The reflexive *himself* is c-commanded by the antecedent *Poirot* within a TP. The *dependency trees* for this sentence and the ungrammatical sentence in (1b) are given in Fig. 3.1 below[4].

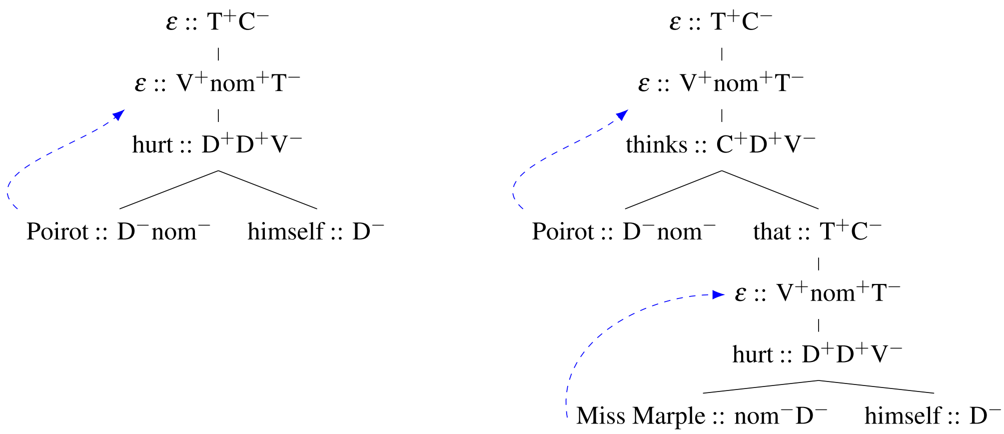


Figure 3.1: Dependency trees for the grammatical *Poirot*$_i$ *hurt himself*$_i$ and the ungrammatical *Poirot*$_i$ *thinks that Miss Marple hurt himself*$_i$

The *c-strings* of the reflexive *himself* are provided in (8) for the monoclausal grammatical phrase and for the ungrammatical sentence, respectively. One can verify that these lists in fact encode the list of c-commanders for *himself* in each of these sentences.

(8) a. *cs*(himself)= himself :: D^- Poirot :: $\mathrm{D}^-\mathrm{nom}^-$ hurt :: $\mathrm{D}^+\mathrm{D}^+\mathrm{V}^-$ ε :: $\mathrm{V}^+\mathrm{nom}^+\mathrm{T}^-$ ε :: $\mathrm{T}^+\mathrm{C}^-$

[4]Note that the trees are minimal in that they do not include functional heads such as *v*. This is immaterial to the analysis and I have not included such heads for simplicity purposes.

b. * cs(himself)= himself :: D^- Miss Marple :: nom^-D^- hurt :: $D^+D^+V^-$ ε :: $V^+nom^+T^-$ that :: $V^+nom^+T^-$ Poirot :: D^-nom^- thinks :: $C^+D^+V^-$ ε :: $V^+nom^+T^-$ ε :: T^+C^-

This format of representing *c-strings* is awfully verbose. Therefore, I am going to omit the features for the time being and only keep the LIs and/or functional heads. This would leave us with the following well-formed and ill-formed *c-strings*.

(9) a. cs(himself)= himself Poirot hurt(V) T C

b. * cs(himself)= himself Miss Marple hurt(V) T that Poirot thinks(V) T C

One can see that in the well-formed case, the reflexive is immediately followed by a potential antecedent in the *c-string*, while in the ill-formed case, some elements intervene between the reflexive and its potential antecedent. One such element is the T-head. If we follow the naive definition of the binding domain, we can say that the *c-strings* in which a T appears between a reflexive and its potential antecedent are ill-formed. Representing the reflexive with the letter *R* and its potential antecedent with the letter *D*, and including the symbol ϕ to show the ϕ-feature match between these two, we say we cannot have the following configuration in (10b). For the time being, I am writing the constraints in a descriptive manner, with the computational analysis following later in §3.4.

(10) a. cs(R)= $R_{[\phi]}...D_{[\phi]}$

b. * cs(R)= $R_{[\phi]}...T...D_{[\phi]}$ (Version 1- To be revised)

(10a) indicates a requirement that a reflexive anaphor should have an antecedent somewhere in its linguistic environment, and (10b) indicates that the binding cannot be established if there is an intervening T-head between the reflexive and its potential antecedent. To clarify, it is easier to interpret the two configurations together, meaning that (10a) is only valid if (10b) is not observed. In other words, the dots in between R and D in (10a) should not instantiate a T head. By the same token, the dots between R and T in (10b) should not instantiate a D head for the configuration to be bad.

This would correctly rule out the sentences in (1b) and in (1c) for that matter, repeated below in (11b) and (11c), respectively. This is because only in (1a) do we have an *R* followed by its antecedent *D* without a T-head in between.

(11) a. [$_{\text{TP}}$ Poirot$_i$ hurt himself$_i$].

b. * Poirot$_i$ thinks [$_{CP}$ that [$_{TP}$ Miss Marple hurt himself$_i$]].

c. * Poirot$_i$ believes [$_{CP}$ that [$_{TP}$ himself$_i$ is the best detective]].

It can also correctly rule out sentences such as (12a) and (12b), in which the reflexive is not c-commanded by the potential antecedent with the matching ϕ features.

(12) a. * I expect himself$_i$ to invite Poirot$_i$.

b. * Poirot$_i$'s sister invited himself$_i$.

c. * Poirot$_i$ believes that himself$_i$ is the best.

d. * Poirot$_i$ believes that himself$_i$ is the best detective.

The *cs*(himself) for these sentences are given below in the same order. The reader is encouraged to draw the *dependency trees* for the sentences and verify the *c-strings*. These *c-strings* are ill-formed.

(13) a. * *cs*(himself)= himself invite T I expect T C $\rightarrow$ $R_{[\phi]}$...

b. * *cs*(himself)= himself 's invited(V) T C $\rightarrow$ $R_{[\phi]}$...

c. * *cs*(himself)= himself is(V) T that Poirot believes(V) T C $\rightarrow$ $R_{[\phi]}$...T...$D_{[\phi]}$

d. * *cs*(himself)= himself is(V) T that Poirot believes(V) T C $\rightarrow$ $R_{[\phi]}$...T...$D_{[\phi]}$

The rules in (10) need to be modified to account for antecedent-less reflexvixes. This can be handled by adding the restriction, **cs*(R)= $R_{[\phi]}$... to our conditions in (10), which would rule out the ungrammatical sentences in (12a) and (12b). My revised rules are shown in (14) below, with the addition of (14b).

(14) a. *cs*(R)= $R_{[\phi]}$...$D_{[\phi]}$

b. * *cs*(R)= $R_{[\phi]}$...

c. * *cs*(R)= $R_{[\phi]}$...T...$D_{[\phi]}$ (Version 2- To be revised)

As the configurations show, the first two sentences in (12) are trivially out because there is no potential antecedent for the reflexive in the list of its c-commanders. The last two sentences, on the other hand, are out because a T intervenes between the reflexive and its potential antecedent in the *c-strings*.

Although this sounds elegant, we already identified cases for which only having a T-head does not block the binding relation to establish, as we saw in (2) and (3), repeated below in (15a) and (15b), respectively. Moreover, we also looked at cases in which other phrase types construct binding domains, such as (4a), repeated here in (15c) for convenience.

(15) a. Poirot$_i$ believes [$_{TP}$ himself$_i$ to be the best]. **ECM**

b. Poirot$_i$ believes [$_{CP}$ that [$_{TP}$ a picture of himself$_i$ will be on show at the exhibition]]. **Picture-phrase**

c. * Poirot$_i$ believes [$_{DP}$ Miss Marple's description of himself$_i$]. **DP Binding Domain**

Let us go through them one by one. Starting with (15a), I previously mentioned that this sentence is an ECM case, with an infinitival embedded clause. I also introduced the notion of SUBJECT and the fact that it corresponds to AGR in finite clauses. To elaborate further, tenseless clauses might not have any form of a *subject*. The SUBJECT is mainly present in finite clauses to encode morphological agreement between the AGR head and the clausal subject. One can propose that specifying tense on the T-head can correctly account for this sentence type. However, not every tenseless or infinitival clause is subject-less *per se*. Take the following sentences as an example.

(16) a. To identify oneself here would be wrong. (Haegeman, 1994: 261)

b. Praising herself bothered Lisa. (Larson, 2009: 341)

c. Wolfgang wants [$_{TP}$ to outdo himself]. (Storoshenko, 2017: 10)

These sentences are all control sentences with a covert PRO acting as the subject of the infinitival clauses. This PRO can act like an overt DP and bind the anaphor in its c-commanding domain.

(17) a. PRO$_i$ to identify oneself$_i$ here would be wrong.

b. PRO$_i$ praising herself$_i$ bothered Lisa.

c. Wolfgang wants [PRO$_i$ to outdo himself$_i$].

To account for the presence of a subject form in the clause, we only need to make our feature system stronger. This can be done by adding two features to the T-head: one that can take care of the finiteness or tense, [FIN], and another that can account for control cases, [CONTROL]. These changes in the T-head are sufficient to capture the data, given that the

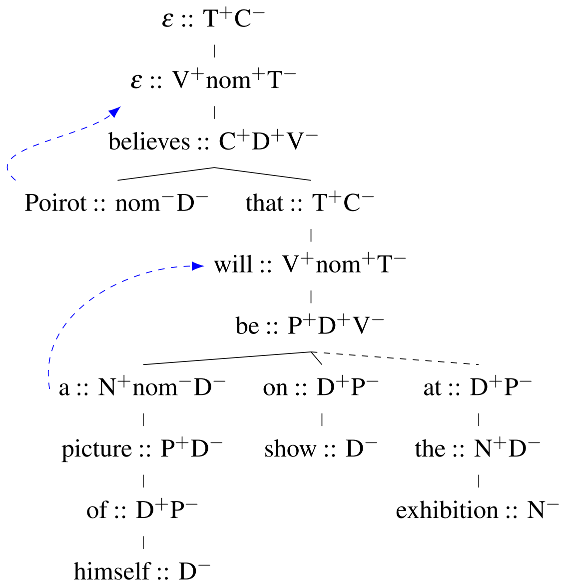


Figure 3.2: Dependency tree representation of *Poirot believes that a picture of himself will be on show at the exhibition.*

PRO can act as a potential antecedent and itself is a DP. That being said, we will have the following configuration for the clausal binding.

(18) a. cs(R)= $R_{[\phi]}...D_{[\phi]}$

b. * cs(R)= $R_{[\phi]}...$

c. * cs(R)= $R_{[\phi]}...T_{[\text{FIN, CONTROL}]}...D_{[\phi]}$ (Version 3- To be revised)

However, this is still unable to rule in the grammatical repeat sentence in (15b). As the reader can confirm, the *c-string* of *himself* in (19) is ill-formed with respect to the constraint I have proposed. It is because there is a finite T intervening between the reflexive and its potential antecedent.

(19) *cs*(himself)= himself of picture a be **will(T)** that(C) Poirot believes T C

As explained in the previous section, this example involves a violation of the *i-within-i* filter or circularity in reference in the sense of Chomsky (1981, 1993). The *dependency tree* for this sentence is provided in Fig. 3.2 for elaboration.

Using indexation for the reflexive, the head of the DP containing it, and the T-head with the AGR feature, we end up having the following *cs* in (20) for the reflexive *himself*.

However, in a formal language theory setting, we would not want to use indices, as it has been shown that "...consistency of theories that employ free-indexation is, in general, undecidable (Rogers, 1998: 62)". That being said, the goal here is to illustrate what *c-strings* are in principle capable of. As far as I can tell, if we just think of binding theory as a mechanism of determining surface forms of pronominal elements — which is what the *subregular* perspective is about — then *i-within-i* doesn't matter.

(20) *cs*(himself)= himself$_i$ of picture a$_i$ be will(T)$_i$ that(C) Poirot believes T C

In order to account for this condition, what we need to do before applying our rules is to first check the first maximal projection containing the reflexive. If the head of the maximal projection and the reflexive are co-referenced, the first T in the *c-strings* should be ignored. And the domain needs to be expanded. For the complexity analysis, I will show that we can model these phrases by modifying our tier projection rules without the need to use indices.

(21) a. $cs(\mathrm{R}) = \mathrm{R}_{[\phi]} ... \mathrm{D}_{[\phi]}$

b. * $cs(\mathrm{R}) = \mathrm{R}_{[\phi]} ...$

c. if $[_{\gamma_i} ... \mathrm{R}_i ...]$:

* $cs(\mathrm{R}) = \mathrm{R}_{[\phi]} ... \mathrm{T}_{[\textsc{fin, control}]} ... \mathrm{T}_{[\textsc{fin, control}]} ... \mathrm{D}_{[\phi]}$

else:

* $cs(\mathrm{R}) = \mathrm{R}_{[\phi]} ... \mathrm{T}_{[\textsc{fin, control}]} ... \mathrm{D}_{[\phi]}$ Version 4- To be revised

Another case that needs to be addresses is the case of DPs with an overt specifier, such as the sentences in (15c), repeated below in (22). Fig. 3.3 shows dependency trees for two well-formed sentences, one with an overt DP specifier (*Poirot believes Miss. Marple's description of himself*) and one without (*Poirot believes any description of himself*). In the sentence with an overt DP specifier, it is the specifier that binds the reflexive, making the DP a binding domain. In contrast, in the sentence with no overt specifier, the binding domain remains TP and the subject of the TP binds the reflexive.

(22) * Poirot$_i$ believes [$_{\mathrm{DP}}$ Miss Marple's description of himself$_i$].

DP Binding Domain

Our conditions can correctly account for the latter case, but not for the former one, since they enforce TP as the binding domain. To enforce DP with specifier as the binding domain

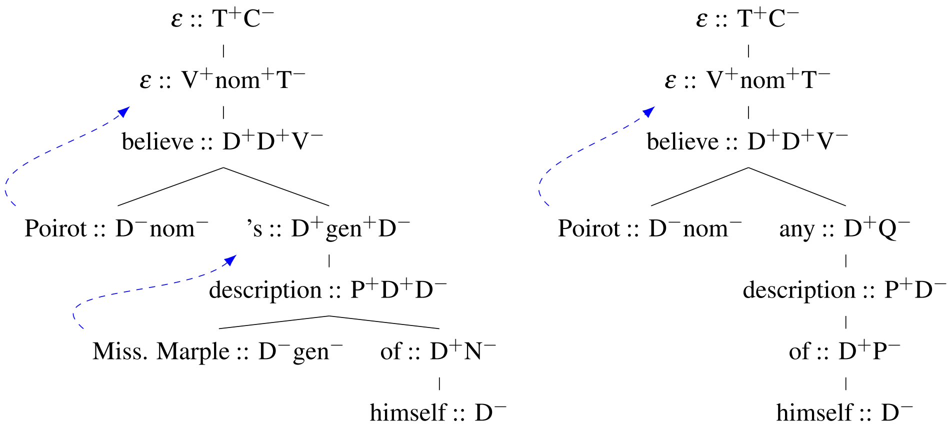


Figure 3.3: DP as binding domain in derived nominals with overt specifier

as well, we need to look at the *c-strings* of the reflexive for such an example. The *c-strings* for $himself$ in the tree in left is as shown in (23).

(23) *cs*(himself)= himself of Poirot description ’s Miss. Marple believe T C

One can say that this falls under the first condition in (21), which predicts a well-formed *c-strings* if there is no intervening T-head between a reflexive and its potential antecedent. However, let us compare this to a minimally different sentence, where Poirot and Miss. Marple switch places. The *c-strings* for the sentence *Poirot believes Miss. Marple’s description of himself* is as (24) shows.

(24) *cs*(himself)= himself of Miss. Marple description ’s Poirot believe T C

In this sentence too, there is no intervening T-head between the reflexive and the potential antecedent with matching ϕ features. No other condition in (21) can correctly accept (23) and rule out (24). We need to add another condition to take into account the overt specifier in the lower DP and enforce that as a binding domain. Comparing the two trees in Fig. 3.3 shows that the D heads in the two differ in their *licensor* features. The D head without an overt specifier does not have one, while the D head with an overt specifier has a gen$^+$ feature on it, which selects another DP to move to its specifier position, *Poirot* in our example. Taking this into account, and adding it to our conditions, I propose (25) as well-formedness conditions for local reflexives. Note that I have also shortened the configurations to exclude the potential antecedent, $D_{[\phi]}$. This is because the configuration can already be ruled out if we find a T-head or a D-head that constitutes a binding domain, after which the need to look for a potential antecedent is needless.

(25) a. * cs(R)= $R_{[\phi]}$...

b. cs(R)= $R_{[\phi]}$...$D_{[\phi]}$

c. * cs(R)= $R_{[\phi]}$...$D_{[gen^{+}]}$

d. if $[_{\gamma_1}$...R_i...]:

cs(R)= $R_{[\phi]}$...$T_{[\text{FIN, CONTROL}]}$

else:

* cs(R)= $R_{[\phi]}$...$T_{[\text{FIN, CONTROL}]}$ Final Version

3.2 Non-local Binding: Cross-linguistic Data

Anaphors are not necessarily bound locally. Scandinavian languages are the most well-known cases which have non-locally bound reflexives. English, also, exhibits this property in some constructions. Long Distance Anaphors (LDA) can be categorized in the following way.

(26) a. They allow an antecedent outside the governing category.

b. Their antecedents are subject to a more restrictive prominence condition than c-command. The most common requirement is that the antecedent must be a subject.

c. LDA are restricted to reflexives. Reciprocals are not allowed as LDA.

d. LDA are morphologically simplex. Morphologically complex anaphors are local.

e. Outside the local domain there is no complementarity between pronouns and anaphors.

(Reuland and Koster, 1991: 10-11)

We can find examples of such LDA in Scandinavian languages, in which the anaphor *sig* can be bound by the matrix subject outside an infinitival clause. This is unlike English where the binding is local and within the lower clause.

(27) a. $Pétur_i$ bað $Jens_j$ um [PRO_j að raka $sig_{i/j}$] **Icelandic**

b. Peter$_i$ bad Jens$_j$ om [PRO$_j$ at barbere sig$_{i/j}$] **Danish**

c. Peter$_i$ asked Jens$_j$ [PRO$_j$ to shave himself$_{*i/j}$] **English**

(Thráinsson, 1991: 51)

The intervening subjects of the infinitives do not seem to have any effect as the examples in (28) show. None of the intervening subjects, i.e. *þig* and *mig*, block the binding relation between the anaphor and its antecedent.

(28) a. Anna$_i$ telur [þig hafa svikið sig$_i$]
Anne believes you (ACC) have (Inf) betrayed self

b. Anne$_i$ hørte [mig snakke med dig om sig$_i$]
Anne heard me talk (Inf) to you about self (Thráinsson, 1991: 51)

This observation has been reported cross-linguistically, for languages like Chinese (29), Gothic (30), Russian (31), Persian (32), among others. This is not to say that the behavior of these reflexives are exactly the same; rather, to show that long-distance binding exists in a variety of language families. Moreover, the goal is to show that having a computational perspective, these theoretically or empirically diverse set of reflexives can be accounted for using a similar mechanism and machinery.

(29) Zhangsan$_i$ renwei [Lisi$_j$ zhidao [ziji$_{i/j}$ de taitai shi yige da hao ren]].
Zhangsan think Lisi know REFL DE wife is one-CL big good man
'Zhangsan$_i$ thought that Lisi$_j$ knew that his own$_{i/j}$ wife was a very good person.'

(Tang, 1989: 45)

(30) Þai$_i$-ei ni wildedun [mik þiudanon ufar sis$_i$/(*im$_i$)]
who$_i$ not they-wanted me to-rule over selves$_i$/*them$_i$
'who didn't want me to rule over them' (reported in Harbert, 1995: 17c)

(31) On$_i$ ne razrešaet mne [PRO$_j$ proizvodit' opyty nad soboj$_{i/j}$]
He$_i$ not allow me$_j$ [PRO$_j$ to-perform experiments on self$_{i/j}$
'He$_i$ does not allow me to perform experiments on himself$_i$/myself$_j$.'

(reported in Rappaport, 1986: 15a)

(32) Sohrāb$_i$ be Āraš$_j$ goft [ke Minā$_k$ hatman bā xod-eš$_{i/j/\#k}$ tamās mi-gire].
Sohrab to Arash said that Mina certainly with self-3SG contact DUR-get
'Sohrab$_i$ said to Arash$_j$ that Mina$_k$ will certainly contact self$_{i/j/\#k}$'

(Abdollahnejad and Storoshenko, 2020, ex.1)

A popular opinion is that this long distance binding is correlated with the nature of the reflexives. More specifically, monomorphemic or simplex reflexives, which do not have a pronominal element or ϕ-features, such as Scandinavian *sig* or Russian *sebe*, can participate in long distance biding, whereas complex reflexives cannot (Pica, 1987; Reuland and Koster, 1991, a.o.). Thráinsson (2017) makes a slightly revised claim that if a language has both simplex and complex reflexives, the complex one typically does not participate in long distance binding.

Another property that has been attributed to simplex reflexive is being subject-oriented, meaning that they tend to be bound only by the subject. The following Russian example illustrates this point.

(33) Ivan$_i$ rasskazal Borisu$_j$ o sebe$_{i/*j}$.
Ivan.NOM told Boris.DAT about self
'Ivan$_i$ told Boris$_j$ about himself$_{i/*j}$.' (Antonenko, 2012: 13)

Compare this example to English examples below, in which the reflexive can be bound by either the subject of the sentence or the object, both direct and indirect.

(34) a. John$_i$ told Bill$_j$ about himself$_{i/j}$.
b. John$_i$ showed Bill$_j$ to himself$_{i/j}$. (Antonenko, 2012: 106-107)

Given the examples up to now, one might posit that the binding domain for such reflexives is larger than English-type ones. Manzini and Wexler (1987) is one such study that proposes cross-linguistic differences can be parametrized by the definition of binding domain or governing category. (35) provides their condition for reflexives like *sig*, where the "referential" Tense is a tense whose properties are inherently defined, such as an indicative tense.

(35) γ is a governing category for α iff γ is the minimal category that contains α and a governor for α and has a "referential" Tense. (Manzini and Wexler, 1987: 417)

This condition, in principle, extends the binding domain beyond the first T-head. Now, we can model this condition using *c-strings*.

3.2.1 Modeling LD Binding: Extended Binding Domain

This section uses *c-strings* to model long-distance binding in the sense of the extended binding domain of Manzini and Wexler (1987), for whom the binding domain for *sig* type reflexives is beyond the first T that contains them. This is in contrast with local reflexives whose binding domain is generally the first TP that contains them.

The *c-strings* for *sig* in the Icelandic example in (28a), repeated in (36) for convenience, are as given in (37).

(36) Anna$_i$ telur [þig hafa svikið sig$_i$]
Anne believes you (ACC) have (Inf) betrayed self (Thráinsson, 1991: 51)

(37) *cs*(sig)= sig þig svikið T C Anna telur T C

Like local reflexives, these long-distance reflexives require an antecedent. Given that these reflexive are generally not locally bound, with the exception of being co-arguments of intrinsically reflexive verbs[5], condition (a) of (25) holds for them too. However, unlike local reflexives for which an intervening T-head renders the *c-strings* illicit, for *sig* type reflexives, it is a requirement. Therefore, we have the following conditions for these reflexives.

(38) **LD Binding Conditions**

[5]These anaphors do not always reject being locally-bound. Consider the well-formed sentences, with locally bound forms of the Dutch reflexives *zich* in (ia & ib), and the Norwegian *seg* in (ic & id).

Referring to Everaert (1986, 1990), Reinhart and Reuland (1993) attribute this difference in the behavior of these LDA to the inherent properties of verbs. They propose that some verbs, such as *schamen* 'shame' are intrinsically reflexive, and so, their predicates are reflexive-marked. Consequently, the anaphors are locally bound. Verbs like *wassen* 'wash', which do allow a distinct object, are listed twice in the lexicon, both as reflexive and as non-reflexive; their reflexive entry allows the LDA, while their transitive entry occurs with a SELF anaphor.

(i) a. Max wast zich. **Dutch**
Max washes self
b. Max schaamt zich.
Max shames SE (Max is ashamed)
c. Jon wasket seg. **Norwegian**
Jon washed self
d. Jon skammer seg.
Jon shames self (Reinhart and Reuland, 1993, ex.18)

a. $* \, cs(\mathrm{R}) = \mathrm{R}_{[\phi]}...$

b. $cs(\mathrm{R}) = \mathrm{R}_{[\phi]}...\mathrm{T}_{[\textsc{referential}]}...\mathrm{D}_{[\phi]}$

A comparison between these rules for long-distance anaphors and the rules for local anaphors in (25), repeated here in (39) for convenience, is in order.

(39) **Local Binding Conditions**

a. $* \, cs(\mathrm{R}) = \mathrm{R}_{[\phi]}...$

b. $cs(\mathrm{R}) = \mathrm{R}_{[\phi]}...\mathrm{D}_{[\phi]}$

c. $* \, cs(\mathrm{R}) = \mathrm{R}_{[\phi]}...\mathrm{D}_{[\mathrm{gen}^{+}]}$

d. if $[_{\gamma_1}...\mathrm{R}_i...]$:
$* \, cs(\mathrm{R}) = \mathrm{R}_{[\phi]}...\mathrm{T}_{[\textsc{fin, control}]}...\mathrm{T}_{[\textsc{fin, control}]}$
else:
$* \, cs(\mathrm{R}) = \mathrm{R}_{[\phi]}...\mathrm{T}_{[\textsc{fin, control}]}$ Final Version

Let us take the simpler cases first. For both anaphor types, not having an antecedent results in ungrammaticality. This is captured by the (a) clause in both of the conditions.[6] The (b) clauses enforce the environments in which these anaphors can appear. For local reflexives, the dependency between the anaphor and its antecedent should not be intervened by an domain-defining element T. While for long-distance anaphors having such an element is crucial. This is the major difference between these two reflexive types. However, as we will see in §3.4.3, they are both subregular as they are both at most IO-TSL.

The next two sections address other accounts of long-distance anaphors and shows how they can both be modeled by my proposed *c-strings*. The goal is to show how different theoretical analyses can be modeled from a complutational perspective and my proposed *c-strings*. In other words, once you adopt *c-strings* as an intermediate repesentation, a variety of proposals fall under the same complexity.

[6]Some logophoric anaphors are exceptions. For instance, there are cases where the anaphor can stand on its own with no potential antecedent in the linguistic environment, such as:

(i) There were five tourists in the room apart from myself. (Reinhart and Reuland, 1993: 669)

3.2.2 LD Binding and LF movement (Pica, 1987)

The LF movement of anaphors was originally proposed by Pica (1987). He proposes that reflexives need to undergo LF movement in order to be *saturated*, in the sense of Higginbotham (1985) who claims that all arguments must be saturated; and if a projection is not saturated at one level, it will have to do so at another level. Again, following Higginbotham (1985), who claims that the head nouns of NPs have an open position, Pica (1987) claims that reflexives are not saturated, as in (40) and so, they need to undergo movement to fulfill this requirement. The differences between simplex and complex reflexives amounts to them being a head or a phrase, respectively. He claims that while simplex reflexives such as Danish *sig* are heads directly dominated by a maximal projection, English reflexives include a head *self* and a specifier, which can be filled by a genitive (e.g. *my*) or an accusative (e.g. *them*). But, in any case, there is an open position that needs to be saturated, hence, the LF movement of anaphors. It is basically the nature of this movement that differentiates between these two groups of anaphors.

(40) An anaphor is an argument which is not saturated at S-structure (Pica, 1987, ex.III)

Since complex anaphors are considered to be NPs, while the simplex ones are considered to be heads, the nature of their LF movement differs. The head anaphors (in Pica (1987)'s term X^0 anaphors) are subject to head movement constraints and the NP anaphors are subject to phrase movement constraints. The first group can attach to heads, whereas the second group can adjoin to other maximal projections. Following Chomsky (1986b), he suggests that head anaphors move to INFL (T) to be interpreted. This explains why such anaphors are always subject oriented, while NP anaphors are not. This contrast can be seen in examples from Danish below. While the head anaphor in (41a) is unable to be bound by the object, the object in (41b) can in fact bind the NP anaphor.

(41) a. * Jeg fortæller Hans$_i$ om sig$_i$
'I tell John about himself.'

b. Jeg fortæller dem$_i$ om hinanden$_i$
'I tell them about each other.' (Pica, 1987, ex 10-11)

In addition, he also proposes that head anaphors can undergo successive cyclic movement to higher INFL's. This proposal comes from the fact that these anaphors can refer to the subject of the matrix clause as well. However, this movement is only available if the movement does not cross a tensed clause. Compare the two Icelandic sentences in (42). While the reflexive can refer to the matrix subject in (42a), it does not show the same property in (42b). This is due to the fact that it needs to cross an indicative clause in the latter, and it is simply impossible to do so.

(42) a. Jón$_i$ sagði þeim$_j$ [að María elski (SUBJ) sig$_{i/*j}$]
'John told them that Mary love himself.'
b. * Jón$_i$ veit [að María elski (IND) sig$_i$]
'John knows that Mary loves himself.' (Pica, 1987, ex 18-19)

One can question the reason why the reflexive cannot be bound by the object in (42b) because, in principle, it is c-commanded by it and therefore, there is nothing preventing the binding relation from being established. Pica (1987) proposes that the head anaphors can undergo INFL to INFL movement using a C head, which is either already empty or its content gets deleted in between the movements to be used as an escape hatch. In other words, for him, the binding relation is established after the reflexive has reached its final position. He explains that the C head in subjunctive or infinitival clauses is empty, since they do not have to have an overt complementizer (as suggested by Ritter and Szabolcsi (1985)). The fact that long distance binding of head reflexives across a subjunctive clause is not possible in Russian (43), where subjunctive inflection is identical with that of an indicative past tense, supports this claim. As a result, *sebja* cannot move to the C position and cannot be bound, and so, that interpretation is excluded.

(43) Vanja$_i$ xočet [čtoby$_k$ vse ljubili sebja$_{*i/k}$]
'Vanja want that everybody love himself.' (Pica, 1987, ex. 20)

3.2.3 Modeling LD Binding using LF movement Account

Let us start with a well-formed sentence:

(44) Jón$_i$ sagði þeim$_j$ [að María elski (SUBJ) sig$_{i/*j}$]
'John told them that Mary love himself.'

If we follow Pica (1987), the reflexive has to move successive cyclically to the matrix INFL (T) position through C. This would give us the following configuration (English words are used):

(45) $[_{\mathrm{CP}}\ [_{\mathrm{TP}}$ John $[_{\mathrm{T}}$ *sig* told them $[_{\mathrm{CP}}\ [_{\mathrm{C}}$ *t* $[_{\mathrm{TP}}$ Mary $[_{\mathrm{T}}$ *t* loves *t*]]]]]]]

To model this, we need to have a move feature in the dependency tree that motivates this type of movement. It is not clear in Pica (1987)'s original work what this motivation is. For simplicity, I am using a ref$^-$ to be the relevant movement feature. The reader recalls that for a movement to occur in the MG mechanism, we need a *licensee* feature on the mover, here the reflexive, and the same *licesor* feature on the goal, here the T. We also know that in the MG mechanism, we do not use intermediary landing sites. The movement happens in one step to the final position. Therefore, we only need to have this probe feature on the final destination. Given this layout, we are left with the dependency tree in Fig. 3.4 for the sentence in (44).

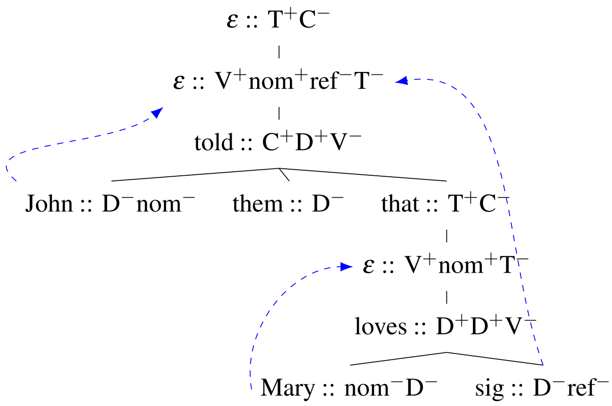


Figure 3.4: Dependency tree capturing the LF movement of the anaphor in Pica (1987)'s terms

Since the mover in the MG system stays in situ, its landing site does not make a difference in the structural relation of the reflexive with other elements. However, the *c-strings* can be more refined to encode this *licensee* feature.

(46) cs(sig)= $\mathrm{sig}_{[\mathrm{ref}^-]}$ Mary loves T C them John told $\mathrm{T}_{[\mathrm{ref}^+]}$ C

Again, the existence of another T-head between the reflexive and its potential antecedent is crucial. We can refine our conditions by using this feature geometry. Remember that the binding across an indicative T is blocked as we saw in (42b). Therefore, our condition should account for this and this feature needs to be encoded, which I do so by using the [IND] feature on the intervening T.

(47) a. * cs(R)= $R_{[\phi]}$...

b. cs(R)= $R_{[\phi,\text{ref}^-]}$...$T_{[\text{IND}]}$...$D_{[\phi]}$

3.3 Binding in Minimalist Era

The analyses of binding in the minimalist syntax can be categorized into two major groups, the doubling constituent theories, e.g. Kayne (2002); Zwart (2002), and the feature-based theories, e.g. Hasegawa (2005); Reuland (2006); Hicks (2009). I give a brief summary of these two theories followed by modeling them in *c-strings* format.

3.3.1 Doubling Constituent Theories of Binding

The general idea in these theories is that a pronoun enters the derivation together with its antecedent. In other words, the pronoun and its antecedent start as a doubling constituent as (48) shows. The antecedent, then, moves out of this doubling constituent to a position where it can get its θ role and Case. I focus on Zwart (2002), however, a similar idea is proposed in Kayne (2002) as well.

(48) $[_{XP}$ [antecedent] [PRONOUN]] (Zwart, 2002, ex.21)

Take the sentence "Himself, John loves" as an example. Zwart (2002) argues that the reason this sentence is interpretable is because *John* and *himself* were adjacent to each other at some point in the derivation (49b), while they had to go through some series of movements, as in (50), to reach their final destinations.

(49) a. Himself, John loves <himself>.

b. [John himself] (Zwart, 2002, ex.8-9)

(50) a. Merge *loves* with *[John himself]*

b. Extract *John* from *[John himself]* and merge it with *loves <John> himself*

c. Extract *himself* and merge it to *John loves < <John> <himself>*

Zwart (2002) argues that some properties naturally fall out of this proposal, which are **asymmetry**, **obligatoriness**, **uniqueness**, **c-command**, **locality**, to name some.

The first property ensures the asymmetrical relation between an anaphor and its antecedent. The fact that we should have "John ...himself" but not "himself ... John" falls from the structure proposed in (48), where PRONOUN is the head of XP. That an anaphor must be obligatorily bound follows on the theory proposed here, since anaphoricity is only acquired by a pronoun when merged with an antecedent as opposed to being a lexical property.

Uniqueness and c-command are also explainable via this proposal. By proposing a binary branching requirement, Zwart (2002) limits the number of antecedents that a pronoun can merge with to one. This means that in a sentence like (51), the anaphor could have only been merged with Bill and therefore can only refer to it.

(51) John heard Bill curse himself. (Zwart, 2002, ex.30)

When it comes to c-command, it is also a direct corollary of the doubling constituent structure proposed in (48). Take the following sentences (52) as an example. The case in (52a) is straightforward. *John* and *himself* are together when they merge with *loves*, after which *John* moves out of this doubling constituent structure to merge with *loved [<John> himself]*.

(52) a. John loves himself.

b. *John's mother loves himself.

In order to derive (52b), with intended coreference of *John* and *himself*, *John* would have to be extracted from (53a) and merged internal to the DP [(_)'s mother] (52b).

(53) a. [[John] [PRONOUN]]

b. [DP _ 's [mother]]

However, this operation cannot happen because it would violate one of the following two conditions. It might violate Chomsky (1993)'s Extension Condition (54) if the DP is already merged to the structure containing *John* at the moment of its extraction out of (53a). If not, it would cause an interarboreal operation by targeting two independent phrase markers if the DP in (53b) has not yet been merged to the structure containing John at the moment of its extraction out of (53a). Zwart (2002) takes this as evidence that such a derivation is disallowed.

(54) Extension Condition: No operation Merge applying to a phrase marker α targets a node dominated by α.

The last property that I would like to address here is the **locality** property. Given that the antecedent of the co-constituent structure needs to move out to get its θ role and Case, and the fact that such positions are A-positions, this movement must be A-movement and therefore, clause-bound. This results in binding being local in two ways. Firstly, anaphoricity is established by merging a PRONOUN with its antecedent in the most local configuration possible, and then, the (obligatory) movement of the antecedent can only be to local positions. One exception is the case of ECM, where two (or more) verbs share a single functional domain, (55), for which the extension of the local domain is needed to account for anaphor binding by an argument of a higher verb, as explained earlier. However, the movement of the antecedent *John* is still local if association of a verb and its functional domain is expressed as "L-relatedness".[7] Taking L-relatedness into account, the maximal functional projection L-related to the matrix verb is the local domain for A-movement.

(55) a. John saw himself kiss Mary.

b. ... dat Jan zichzelf Marie zag kussen **Dutch**
... that John himself Mary saw kiss-INF (Zwart, 2002, ex.36)

If we follow doubling constituent account of binding, Principle A of the classical binding theory is deduced from the derivational approach to binding, involving merger of binder and bindee in conjunction with a locality condition on the movement of the antecedent.

This modeling is very trivial in terms of *c-strings*. The only difference with (48) is that the order is the opposite. There is nothing more to it. The only condition is that the antecedent and reflexive should be adjacent, i.e. *R D* with nothing in between them.

3.3.2 Binding and Feature-based Theories

This group of analyses relies on the locality of Agree operations (in the sense of Pesetsky and Torrego (2007)) and Phase Impenetrability Condition (PIC), defined below.

(56) **Agree: Feature Sharing Version**

[7] L-relatedness reflects whether a head carries an L-feature, defined as a lexical feature of the verb. In this view, heads like Tense and Agr are taken to host L-features, but not C.

a. An unvalued feature F (a probe) on a head H at syntactic location α (F_α) scans its c-command domain for another instance of F (goal) at location β (F_β) with which to agree.

b. Replace F_α with F_β, so that the same feature in present in both locations.

(Pesetsky and Torrego, 2007: 268)

(57) Phase Impenetrability Condition (PIC): "In phase α with head H, the domain of H is not accessible to operations outside α, only H and its edge [its specifier(s)] are accessible to such operations." (Chomsky, 2000)

I focus on one such account, presented in Hicks (2009). Hicks (2009) proposes an Operator [OP] Variable [VAR] relation between the reflexive and its antecedent coupled with other assumptions. Firstly, he assumes that complex reflexives consist of a pronominal D head and a nominal *self* complement, as in (58). Additionally, he also argues for the bi-directionality in the probe-goal relation, meaning that the search space for a probe extending 'upwards' if no goal can be found in its c-command domain. Moreover, he proposes that phases are indeed the binding domains for anaphors, as per Lee-Schoenfeld (2004).[8] Lastly, he proposes that an anaphor has an unvalued variable feature when entering the derivation.

(58)
DP
D NP
him self

Using these assumptions, in the sentence in (59), the head D of the anaphor, *him*, merges with NP, *self*. Upon this merge, the [VAR: _] on D unsuccessfully probes NP, and therefore, the search space for the probe extends. Since *John*, a c-commanding element with matching [VAR] feature, enters the derivation before a phase, i.e. *v*P, is built, this probe's unvalued feature is valued. The result is a grammatical sentence.

(59) $[_{TP}\ \text{John}_{<\text{Var:x}>}\ [_{vP}\ <\text{John}_{<\text{Var:x}>}>\ \text{likes}\ [_{VP}\ \text{himself}_{<\text{Var:\_}>}]]]$ (Hicks, 2009: 128)

[8] Lee-Schoenfeld (2004) proposes the following condition for reflexives:

(i) A reflexive must be bound within the minimal phase containing it. (Lee-Schoenfeld, 2004, ex.51a)

This account can correctly rule out sentences where the reflexive is not in enough proximity to its antecedent, e.g. in biclausal cases such as "*John said that Mary likes himself". In this case, the potential antecedent does not enter the derivation within the same *v*P phase as the anaphor and therefore, the anaphor's unvalued [VAR:_] feature cannot be checked and valued.

There are two cases that I would like to discuss here since we have seen their examples throughout the chapter. One is the case of ECM constructions, and the other is the case of picture-phrases. Let us start with the ECM cases.

It is generally believed that the complement of an ECM verb is a TP, as (60) shows. Therefore, the point where the anaphor enters the derivation is crucial in it being licensed.

(60) John believes [$_{\text{TP}}$ himself to love Mary]] (Hicks, 2009: 130)

Hicks (2009) proposes that the embedded subject, *himself*, enters the derivation as the specifier of the lower most *v*P. Since it is at the edge of a phase, according to PIC, it is accessible to a higher clause. Now, it can move to the specifier of the embedded TP clause and get valued by the matrix subject upon its merge, since TP does not constitute a phase.

(61) [<$_{vP}$ John$_{<\text{Var:x}>}$> believes himself$_{<\text{Var:_}>}$ to [$_{vP}$ <himself$_{<\text{Var:_}>}$> love Mary]]
(Hicks, 2009: 130)

The next case is the case of picture-phrases, which can be bound non-locally. Hicks (2009) elaborates that three possible configurations could exist for picture-phrase, (62). These sentences can participate in local binding with a DP-internal PRO, as in (62b). The case of (62c) depicts a logophoric reading for such anaphors, while (62a) shows another possibility that these anaphors have a larger domain.

(62) a. John$_{[\text{VAR: x}]}$ likes [$_{\text{DP}}$ pictures of himself$_{[\text{VAR:_}]}$]
b. John$_{[\text{Var:x}]}$ likes [$_{\text{DP}}$ PRO$_{[\text{Var:x}]}$ pictures of himself$_{[\text{Var:_}]}$]
c. John$_{[\text{Var:x}]}$ likes [$_{\text{DP}}$ pictures of himself$_{[\text{Var:x}]}$]

Hicks (2009) takes another approach and argues that while DPs do not constitute phases, these phrases contain *n*Ps that do form phases. Furthermore, it is *n*Ps with overt subject that constitute phases, and subjectless *n*Ps (*n*Ps with no overt subject or covert PRO) do not. Let us take a look at an example with an *n*P without a subject, (63).

(63) a. John likes [$_{\text{DP}}$ pictures of himself].

b. [$_{v\text{P}}$ John$_{\text{[Var:x]}}$ likes [$_{\text{DP}}$ [$_{n\text{P}}$ pictures of himself$_{\text{[Var:_]}}$]]]. (Hicks, 2009: 155)

Now, compare this to an *n*P with a subject, (64). In this example, the only compatible candidate to value the [VAR: _] of the anaphor is Bill, since the *n*P is a phase. The second potential antecedent *John* is beyond this phase, and therefore, Bill is the entity that binds the anaphor. In the previous example, however, the subjectless *n*P is not a phrase, and neither is the DP. Therefore, the matrix subject, *John* binds the anaphor.

(64) [$_{\text{TP}}$ John [$_{v\text{P}}$ <John> likes [$_{\text{DP}}$ [$_{\text{NumP}}$ Bill's [$_{n\text{P}}$ <Bill$_{\text{[Var:y]}}$> pictures of himself$_{\text{[Var:_]}}$]]]]]. (Hicks, 2009: 154)

The nature of long-distance reflexive such as Dutch *zich* is proposed to be different from complex reflexive such as *zichzelf* or English *self* phrases. Hicks (2009) adopts the proposal by Reinhart and Reuland (1993) that these SE anaphors are D heads, as opposed to the SELF anaphors which are of category N and are selected by a D, as in (58). Moreover, he assumes that these reflexives enter the derivation with a [reflexive] feature and therefore, reflexivize the verbs they merge with. This ensures that such anaphors will automatically be locally bound without appealing to valuation of their [Var] feature. In other words, they do not look for antecedents, unlike the SELF anaphors.

3.3.3 Modeling Minimalist Accounts of Binding

The two approaches explained in the previous two sections can easily be accounted for using *c-strings*. Since the doubling constituent theory is pretty trivial when converted to *c-strings*, as it only requires the reflexive and its antecedent to be adjacent to each other, I focus on the feature-based account.

This account is very similar to the classical binding theory. It involves "domain" in a sense, however, the definition of "domain" has changed a bit and is translated to "phase". Fig. 3.5 shows dependency trees for a simple finite clause, as well as an ECM clause[9]. Notice that the trees are refined to include Hicks (2009)'s proposed structure for SELF anaphors, in (58). They also include a *v* head, as it is crucial in defining phases[10].

[9] I am using the epp$^-$ and epp$^+$ feature to motivate the movement of embedded subject in the ECM case. The EPP feature requires that all clauses must have a subject (Hornstein et al., 2005).

[10] Note that *v* is specified using lower case, however, it should not be mistaken for a licensee/licensor feature. Such features are not italicized. The same holds for *n* as well.

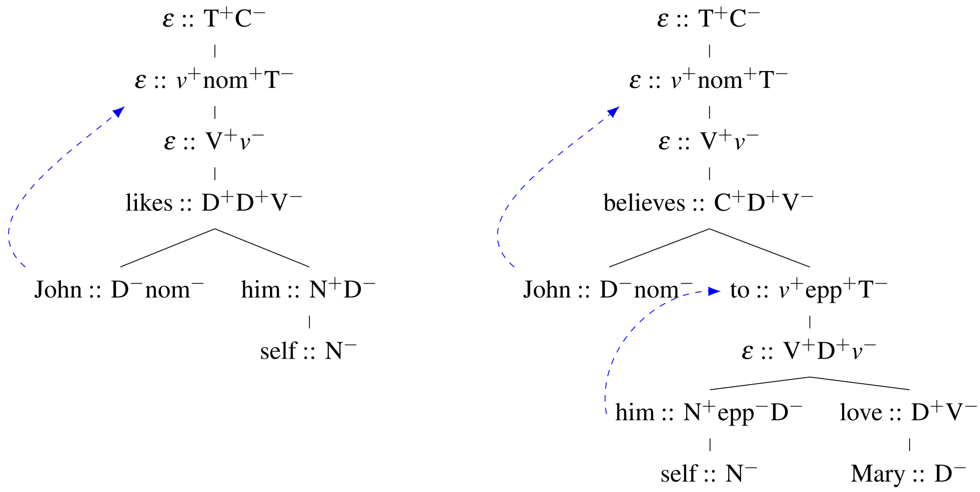


Figure 3.5: Dependency tree representation for a mono-clause (left) and an ECM case (right) using Hicks (2009)'s format for reflexives.

Beginning with the monoclausal case in left, the *c-string* for *self* in the monoclausal example is provided in (65).

(65) *cs*(self)= self him John V *v* T C

Taking Hicks (2009)'s proposal, that reflexive binding should be within a phase and *v* is such a phase, into account, we can conclude that the element which might act as a blocker between a reflexive and its antecedent is the *v*. Therefore, I propose the following forbidden *c-string* .

(66) **cs*(self)= self...*v*...D

This is very similar to the condition I proposed in (10) for classical binding theory. However, it needs revision just like (10) did. Looking at an ECM case would give us more insight. The *c-string* for *self* for such a case is given below:

(67) *cs*(self)= self him V *v* T John V *v* T C

One can see that this *c-string* violates the condition proposed in (66), as there is a *v* head between the reflexive and its antecedent. Therefore, we need to revise our condition or add to it. Remember that Hicks (2009) argues in ECM cases, the embedded subject, here the reflexive, starts off at edge of the lower *v*P where it is visible to the higher phase, rather than the specifier the VP. Then, it moves to the specifier of TP to fulfill the requirements on the T-head, which I show here by a epp$^+$ feature on the T-head.

Notice that the *c-string* for the reflexive in this structure is ill-formed according to the configuration in (66), because the *v* that selects it constructs a phase and therefore is a blocker for binding relations to be established. We need to specify or encode the fact that this reflexive is at the edge of the *v*P and therefore, the first *v*P or the one that contains it does not act as a barrier.

The licensee feature on the reflexive comes to the rescue here. Note that the reflexive in the ECM example is inside a DP, headed by *him*, with a epp$^-$ feature. This feature indicates that the phrase containing *self* will eventually move to a position where it will be visible to higher phrases. This also indicates that this phrase is an ECM embedded subject that needs to move to T to satisfy its requirement. Therefore, we need to include this particular feature in our *c-string*, as (68) shows.

(68) *cs*(self)= self him :: nom$^-$ V *v* T John V *v* T C

The condition on *c-strings*, then, can specify that in such cases, the first *v* is not a barrier. Note that they can be shortened to exclude the antecedent D, in parallel with the shortening that I implemented in the classical binding cases.

(69) a. **cs*(self)= self...*v*...*v* if *self* is selected by an element with nom$^-$ feature

b. **cs*(self)= self...*v* elsewhere

The next thing that needs to be addressed in the case of picture-phrases. Hicks (2009) argues that *n* constitutes a phase when it has an over specifier. The structure of such phrase, following his proposals, is given in Fig. 3.6.

This is also very similar to the cases reported in (4), repeated below for convenience.

(70) a. * Poirot$_i$ believes [$_{\text{DP}}$ Miss Marple's description of himself$_i$].

b. Miss Marple believes [$_{\text{DP}}$ Poirot$_i$'s description of himself$_i$].

c. Poirot$_i$ believes [$_{\text{DP}}$ any description of himself$_i$]. (Haegeman, 1994: 213-214)

The proposal that *n* with an overt specifier acts as a phase and consequently, a binding domain, is evident from having a DP with a gen$^-$ feature. This can be encoded using *c-strings* in the following way. We need this specification to distinguish such cases with subject-less *n*Ps.

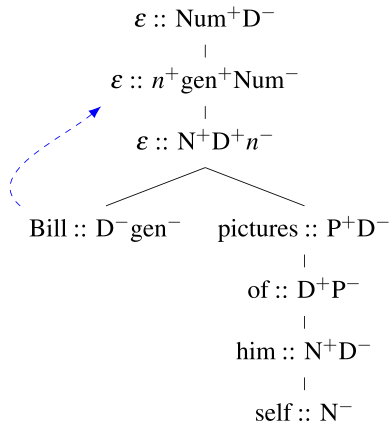


Figure 3.6: Dependency tree representation for the internal structure of *n*P in Hicks (2009) form

(71) *cs(self)= self...D :: gen$^+$...D

3.3.4 Interim Summary

We have looked at various binding theories that address anaphor binding conditions, showing that they can all be encoded using *c-strings*. We started with classical binding theory §3.1, followed by non-local binding cases in §3.2. The section after that covered some of the more recent theories of binding in the minimalist framework §3.3.

Before moving further, it is worth noting that the *c-strings* we have so far looked at provide the list of c-commanders that contain the reflexive. There is a difference between a containing c-commander and a non-containing one. A T-head only marks the edge of a binding domain if it is a containing c-commander. Put it another way, a c-commanding T-head that projects a TP that does not contain reflexive X does not mark the edge of X's binding domain. Suppose we have a language where TPs can be direct objects. If we do not differentiate between a containing c-commanding T head and the non-containing one, we would predict that [Mary explained TP to herself] would be illicit because the *c-string* of herself would be "herself to T explained John T C", which violates the tier constraint against {RT}. On the other hand, the opposite holds for D heads, that is a c-commanding D-head that projects a DP containing a reflexive is not a possible binder of that reflexive, e.g. "*a picture of itself".

In order to account for this idiosyncrasy and to differentiate a containing c-commander from a non-containing one, I assume that for any given head H, the *c-strings* have separate symbols for containing c-commander and non-containing c-commander for a binding domain head. We can use a [+CON] and [-CON] to show this difference. Using this format, the need to use indices for picture-phrases is eliminated. Let us consider the example in (20), repeated here in (72).

(72) *cs*(himself)= himself$_i$ of picture a$_i$ be will(T)$_i$ that(C) Poirot believes T C

If we use the distinctive feature I proposed here, i.e. [CON], having in mind that a containing D head does not act as a binding domain, and so, is not required to be considered, I can model the picture-phrases as the following, and get rid of indices. This would leave me with the condition in (74) on *c-strings* for picture-phrases.

(73) *cs*(himself)= himself of picture $\mathrm{a}_{[+\textsc{con}]}$ be will(T)$_i$ that(C) Poirot believes T C

(74) *cs*(R)= $\mathrm{R}_{[\phi]}...\mathrm{T}_{[\textsc{fin, control}]}$, if $\mathrm{R}...\mathrm{D}_{[+\textsc{con}]}$

Note that since most cases in the thesis include containing Ts and non-containing Ds, I do not make use of this feature throughout the thesis to avoid unnecessary long *c-strings*. But, if need arises, this feature can be used to differentiate between the two. Table 3.1 summarizes the conditions for *c-strings* for various theories.

Absolute Condition	**cs*(R)= $\mathrm{R}_{[\phi]}...$
Classical Binding	a. *cs*(R)= $\mathrm{R}_{[\phi]}...\mathrm{D}_{[\phi]}$ b. *cs*(R)= $\mathrm{R}_{[\phi]}...\mathrm{T}_{[\textsc{fin, control}]}$, if $\mathrm{R}...\mathrm{D}_{[+\textsc{con}]}$ c. **cs*(R)= $\mathrm{R}_{[\phi]}...\mathrm{T}_{[\textsc{fin, control}]}$, elsewhere d. **cs*(R)= $\mathrm{R}_{[\phi]}...\mathrm{D}_{[\mathrm{gen}^+]}$
Non-Local Binding	a. Extended Binding Domain: *cs*(R)= $\mathrm{R}_{[\phi]}...\mathrm{T}_{[\textsc{referential}]}...\mathrm{D}_{[\phi]}$ b. LF Movement: *cs*(R)= $\mathrm{R}_{[\phi,\mathrm{ref}^-]}...\mathrm{T}_{[\textsc{indicative}]}...\mathrm{D}_{[\phi]}$
Minimalist Accounts	a. **cs*(self)= self...*v*...*v* if *self* is selected by an LI with a epp^- b. **cs*(self)= self...*v* elsewhere c. **cs*(self)= self...D :: gen^+

Table 3.1: Conditions for *c-strings* in Different Theoreis

We are finally in a position to discover the computational complexity of these *c-strings*. Next sections address this point showing that binding conditions are reduced to a *subregular* level given the representation proposed in this chapter. This is the first step in showing that syntactic dependencies are less complex that largely assumed, which is the main goal of this thesis. No theory of binding is complete without addressing the interaction of binding and movement. This is the topic for §3.5, where I show how deriving *c-strings* to account for movement gets more complicated; nonetheless, their computational complexity remains the same.

3.4 Subregular Complexity: TSL

We have seen that various binding conditions can be accounted for using *c-strings*. This modeling allows us to represent different approaches to the binding phenomena with a single mechanism, regardless of how different these approaches might be. The next step is to define the computational complexity of these *c-strings*. It is shown that these *c-strings* belong to the *subregular* class of the Input Output Tier-based Strictly Local (IO-TSL) class, an extension of the TSL class. We are already familiar with the TSL class from §2.3.3. However, before moving forward with analyzing the complexity of *c-strings*, I provide a short recap of the key points for TSL. Following that, I define an extension of the TSL class called IOTSL (Input-Output TSL) class, which is the class that binding relations fit into. If the idea of TSL class is fresh in your mind, you can skip to §3.4.2.

3.4.1 TSL- Recap

I provide a quick and intuitive recap of the TSL class to refresh the readers minds before diving into determining the complexity of the *c-strings*.

The class of TSL languages, originally defined in Heinz et al. (2011), is a means of making non-local dependencies local by putting the relevant elements on a tier. This mechanism can be used to model non-local dependencies with blocking effects.

Examples of long-distance phonological dependencies that are TSL include various harmonies, such as sibilant harmony in Berber languages. For instance, long distance sibilant voicing assimilation in Tamajaq Tuareg of the Berber language family results in the complete identity between the sibilant in the prefix and the triggering sibilant in the root that the prefix attaches to (Hansson, 2010). Given the fact that the underlying form for the prefix is [s], and using a ban on non-agreeing sibilants on a sibilant tier, e.g. **sz* or **sʒ*, TSL can capture this harmony process.

TSL is also able to capture long-distance dependencies with blocking effects. For instance, lateral dissimilation in Latin changes [l] to [r] in the suffix *-alis* if it is preceded by [l], but dissimilation is blocked if there is another [r] between the target and the trigger. We saw that this blocking effect can be accounted for by projecting all the laterals on the lateral tier, and ban the {ll} sequence on the tier. This will correctly rule out /lunalis/ because the lateral tier would include the forbidden {ll} factor, while accepting the [floralis] as the lateral tier will not have the two *l* sounds adjacent to each other.

Although TSL can account for various long distance dependencies with or without blocking effects, there are circumstances where TSL alone falls short of accounting for the data. For instance, I showed that sibilant harmony with blocking effects in Imdlawn Tashlhiyt is MTSL where the voicing and anteriority harmony are evaluated separately on two different tiers. Other examples include the RHOL cases in Eastern Cheremis and Dongolese Numbian, which Baek (2017) discusses and shows that TSL is not sufficient to capture the data and needs to take structural features into account. Another example is De Santo and Graf (2019), which addresses a variety of phenomena including sibilant harmony in Samala arguing for Structure-Sensitive TSL (SS-TSL). Graf and Mayer (2018) uses Sanskrit *n*-retroflexion as yet another empirical evidence to argue for an extension of TSL which takes into account both input and output, hence IO-TSL, which I discuss in the coming section. I later show that binding conditions are (at most) IO-TSL using the *c-strings* representation that I have proposed in Chapter 2.

3.4.2 Input Output TSL (IO-TSL)

The power of TSL can be increased by changing the nature of the tier projection π, and making it an input-output strictly local transduction. In IO-TSL dependencies, the tier projection function has to consider the local context in the string, and which symbols are already on the tier. This means that the projection of a symbol on the tier is dependent on its position in the input as well as the existing symbols on the tier. The IO-TSL class is defined by Graf and Mayer (2018), and is an extension of the class TSL-k defined in Heinz et al. (2011), which is identical to IO-TSL-$(1,1,k)$. This shows that IO-TSL is indeed a generalization of TSL. In fact, I-TSL and O-TSL have been independently proposed in computational phonology (Baek, 2017; De Santo and Graf, 2017; Mayer and Major, 2018; Yang, 2018), and Graf and Mayer (2018) show that the combination of the two into IO-

TSL furnishes the additional power that is required for Sanskrit *n*-retroflexion. For a formal definition of the IO-TSL class, please refer to §A.4.

Under the assumption that dependencies in phonology and syntax are of comparable complexity, IO-TSL is a natural candidate for a tighter upper bound on the complexity of constraints on *c-strings*, explained in §2.2.

Example 13. Sanskrit n-retroflexion is IO-TSL: Sanskrit n-retroflexion or *nati* is a combination of various processes, including long-distance assimilation, blocking by preceding coronals, mandatory adjacency to sonorants, blocking by preceding plosives, and blocking by following retroflexes. Please refer to Graf and Mayer (2018) and references therein for more substantive discussion of each of these processes.

1. **Long-distance assimilation**: underlyingly anterior /n/ becomes retroflex [ɳ] when it is preceded in the word by a non-lateral retroflex continuant (/ɾ/, /ɾ̩/, /ɾ̩ː/, or /ʂ/). Examples of this assimilation process are [naɾ-eː **ɳ**a] 'by man', [manuʂj-eː**ɳ**a] 'by human', which can be compared to examples without a non-lateral retroflex for which the /n/ sound surfaces as [n], as in [káːm-eː**n**a] 'by desire', or [joːg-eː**n**a] 'by means.

2. **Coronal blocking:** Coronals that intervene between the trigger and target block *nati*, with the exception of the palatal glide /j/. Examples of this blocking effect can be seen in [**ɾ**á**t**ʰ-eː**n**a] 'by chariot', or [ga**ɾ**u**d**-eː**n**a] 'by Garuda'.

3. **Sonorant adjacency:** The /n/ target must be immediately followed by a non-liquid sonorant. More precisely, the following symbol must be a vowel, a glide, /m/, or /n/ itself. We saw two examples above in process 1, i.e. [naɾ-eː**ɳ**a] and [manuʂj-eː**ɳ**a], where the following sound was a vowel. An example like [ca**ɾ**-a-**n**-**t**i] 'wander (3Pl)' shows how *nati* is blocked, when the following sound is not a non-liquid sonorant.

4. **Velar/labial blocking:** Preceding velar and labial plosives can block *nati* when they occur immediately before the target /n/; and there is a left root boundary, shown by the symbol $\surd$, between target and trigger. Examples of such blocking can be seen in [p**ɾ**-$\surd$aː**p**-**n**oː-ti] 'attains (3s)', and [p**ɾ**a-$\surd$bʱa**g**-**n**a] 'broken'. Comparing them to examples such as [$\surd$**ɾug**-**ɳ**á] 'break(PP)', where one of these conditions is not met, i.e. absence of a root boundary, results in nati, makes this point more clear.

5. **Retroflex blocking:** *nati* is blocked when a retroflex occurs to the right of the target, only when there is a left root boundary between target and trigger. Another condition is the absence of coronal between /n/ and the blocking retroflex, meaning that coronals block this blocking. This blocking is observed in words like [p**ɾ**a-√**n**aks-] 'approach', and [p**ɾ**a-√**ɾ**t-] 'dance forth'. Presence of a coronal blocks this blocking effect as in words like [p**ɾ**a-√**n**eɧ-**t****ɾ**] 'leader', and [√p**ɾ**-**n**a-k-**s**i] 'unite (2s)'.

Graf and Mayer (2018) analyze *nati* by an IO-TSL grammar. Using *R* to represent *R*etroflex triggers, *S* for non-liquid *S*onorant, *C* for *C*oronals and *P* for labial and velar *P*losives, the IO-TSL grammar for Sanskrit n-retroflexion is done following the projection conditions below. Once the tiers are ready, we ban the two factors of *RS* and √*SX (where X is* $\ltimes$*, C or S)*.

- Project every *R*
- Project *S* if it is immediately preceded by [n] in the input
- Project √ if the previous symbol on the tier is *R*
- project *P* if the previous symbol on the tier is √, and the next two input symbols are [n] and *S*
- Project *C* if the previous symbol on the tier is one of *R* or √ or *S*, unless *C* is [n] and the next input symbol is *S*

Let's walk through an example here, starting with the good one on the left in Fig. 3.7. For this example, the *R* symbols, i.e. [n] and [ɾ], are projected on the tier. The *S*, i.e. [a] is projected as well since it immediately follows [n] in the input. There is no √ symbol, hence no projection. As for the *P* symbol, since there is no √ symbol on the tier, there is no projection. Therefore, we get *aɾn*$\ltimes$ on the tier. None of the forbidden factors mentioned above, namely *RS* or √*SX (where X is* $\ltimes$*, C or S)*, are observed on the tier. Consequently, the word is correctly accepted.

As for the ill-formed example, we only have one retroflex, [ɾ], which we project on the tier. The two instances of [a] to represent our *S* symbol are also projected, since they are both immediately preceded by [n] in the input. Lack of projection for √ or *P* is due to lack of √ in the input. As a result, we are left with the *aɾa*$\ltimes$ symbols on the tier. For this example, we can instantiate one of the forbidden factors above, namely the *RS* factor. This is instantiated by the *ɾa* sequence on the tier, marked by the solid blue square in the figure.

Figure 3.7: Sanskrit n-retroflexion process is IO-TSL

In the next section, I show that over *c-strings*, IO-TSL is an upper bound for binding conditions. Therefore, these conditions belong to the *subregular* classes of languages. Remember that a grammar is IOTSL if the tier projection is dependent on the symbols in the input or the output. In other words, we are in the IOTSL class if the tier projection conditions contain sentences such as “project X if it immediately precedes/folllows a Y on the input” or “project X if the previously projected symbol is Y”.

3.4.3 Binding is IO-TSL

Long-distance (syntactic) dependencies all share a common property, which is the fact that there is no upper bound on the length of the dependency. Similarly, there is no upper bound in the length of the *c-strings*. However, as we have seen so far, for various dependencies, only a finitely bounded number of elements actually matter. However, for *c-strings*, we need to take the context of the input and the tier into account as well. This is needed to define which elements matter and should be on the tier. This definition of *relevant* elements depends only on the local context and which symbols are already on the tier. This allows for a very general IO-TSL strategy: construct tiers such that they only contain the relevant elements. Then there are only finitely many distinct tier configurations, and hence the set of well-formed tiers is finite. As for every finite string language SL-*k* for some *k*, separating the well-formed tiers from the ill-formed ones becomes trivial. Therefore, the central challenge for an IO-TSL treatment of c-command dependencies lies in the construction of tiers, not the constraints on those tiers.

The tier construction process also follows a general template. We project tiers from left to right, and the very first element is always projected — this is an input-sensitive projection step. We then use an output-sensitive projection strategy to only project relevant elements.

Example 14. Local Reflexive: Recall the conditions on *c-strings* from (25) for classical binding theory. The most general condition specifies that the dependency between a reflexive and its antecedent cannot be interrupted by a finite T-head, as in "*cs(R)= $R_{[\phi]}...T_{[\text{FIN, CONTROL}]}...D_{[\phi]}$". IO-TSL can enforce this requirement in the following way.

If the first element of a *c-string* is a reflexive that is subject to principle A, only (finite or control) T-heads and matching D-heads need to be projected. Again nothing is projected after this second projection step as principle A is already satisfied or violated depending on whether the second symbol is T or $D[\phi]$. The tier then has one of the following three forms: *reflexive D*, *reflexive T*, or just *reflexive*. The first one is well-formed, the other two ill-formed, as Fig. 3.8 shows.

Let us walk through the mechanism in steps, using two examples, one well-formed and one ill-formed, as in (75), repeated from (9).

(75) a. cs(himself)= himself Poirot hurt(V) T C

b. * cs(himself)= himself Miss Marple hurt(V) T that Poirot thinks(V) T C

The first step is projecting the first element of the *c-strings* on the tier, *himself* in both cases. The second step is projecting the T-head or the D head with matching ϕ features. This leaves us with a *{himself Poirot}*, $R[\phi]...D[\phi]$ for the well-formed sentence and *{himself T}*, $R...T$, for the ill-formed sentence. These factors are enough to accept and reject the well-formed and ill-formed sentences, respectively. This is due to our definition of licit and illicit factors above.

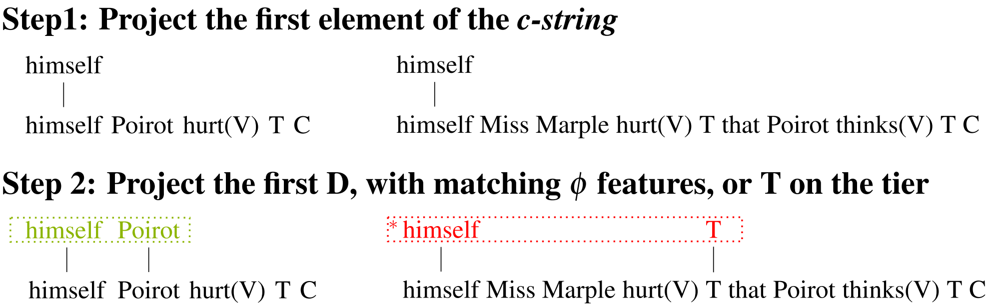


Figure 3.8: IOTSL analysis of Local Binding

Example 15. Long-distance reflexives, like *sig*, can also be accounted for using a similar IO-TSL mechanism. Remember that for such reflexives, the presence of a T-head is required between the reflexive and its antecedent, rather than being forbidden as in the case of local reflexives. The condition "*cs*(R)= $R_{[\phi]}$...T...$D_{[\phi]}$" shows how a well-formed *c-string* would look like in such cases.

To capture this requirement using IO-TSL, we need to project the reflexive *sig*, and a following T and a D head with matching ϕ features. In other words, unlike the case of local reflexives, where we project either the T or D head, in the case of *sig*, we need to actually project a T-head that is following the reflexive and then a D head that is following T.

Figure 3.9: Tier projection for *sig*

Fig. 3.9 shows a good tier for *sig* and a bad one (for simplicity, English words are used). The good tier, as shown in the left, includes the reflexive *sig*, T and antecedent D. The bad tier, on the other hand, includes *sig* followed immediately by a potential antecedent. Although this tier is well-formed for local reflexives, such as English *-self* reflexives, it is not licit for long-distance reflexives[11].

Example 16. Picture-phrases act very similarly to long-distance reflexives. Recall from example (3), that these phrases fall under Chomsky (1981, 1993)'s *i-within-i* filter or circularity in reference. Such phrases extend their binding domain by one clause. This makes them comparable to long-distance anaphors, for which the first T is transparent. In other words, picture-phrases also require an intervening T to be licensed by an antecedent in a higher clause. Just like local reflexives, we project the reflexive on the tier, and the D with matching ϕ features as well as the T-head. It is our factor that would differ for these phrases. If the reflexive is selected by a co-indexing D, our factor is as of the non-local binding cases.

Example 17. DPs as binding domain also fall under the IO-TSL class. For such cases, we project the reflexive on the tier, and every D head that has a gen^{+} feature. We also project the first D with matching ϕ features. The configuration for these cases is "**cs*(self)=

[11] For inherently reflexive verbs, such as "schaamt" in *Max schaamt zich* "Max shames self ", this is licit.

himself T Poirot

himself$_i$ of picture a$_i$ be will(T)$_i$ that(C) Poirot believes T C

Figure 3.10: Tier projection for picture-phrases

self...D :: gen$^+$...D" in the minimalist account, which can be captured using the $\{$self gen$^+\}$ forbidden TSL factor.

Compare the following two diagrams in Fig. 3.11 and Fig. 3.12 for a good and bad tier projection for genitive phrases. While in the good one, the first D acts as a binder, in the second one, the D with gen$^+$ acts as a blocker, and prevents John from binding *self*. This is elegantly captured by IO-TSL as shown in the diagrams.

self Billl

self him of pictures Bill n Num D John likes T C

Figure 3.11: A good tier projection for DPs as binding domains

self Num$_{[\text{gen}+]}$

self him of pictures Mary n Num$_{[\text{gen}+]}$ D John likes T C

Figure 3.12: A bad tier projection for DPs as binding domains

So far, we have seen that we project special elements, e.g. T, D with matchin ϕ features or D$_{[\text{gen}+]}$, on the tier if the first symbol on the tier is a reflexive. If the first symbol on the tier is not subject to any constraints, then nothing else is projected. Proceeding in this fashion, we can define a single IO-TSL grammar that generates every well-formed *c-string* while forbidding those that violate one of our string patterns. This shows that the whole system of syntactic dependencies is IO-TSL, not just each individual dependency.

Each dependency requires only a finite number of elements on the tier, and by considering the entire tier built so far one can ensure that no extraneous material is projected. In addition, each LI can only be subject to finitely many licensing conditions. These two facts jointly entail that the length of tiers can always be finitely bounded, which makes it trivial to provide an SL grammar that rules out all illicit tiers. It is because of this fixed bound on the number of elements that matter for any given c-command dependency that IO-TSL is a safe upper bound on the complexity of syntactic dependencies over *c-strings*. This is also the reason why we contend that IO-TSL could accommodate empirically more adequate characterizations of the phenomena in this chapter — the number of relevant lexical items would still be finitely bounded within any given *c-string*.

Admittedly, this strategy comes at the potential cost of large contexts. Fairly small contexts seem to suffice for realistic examples, though, a smarter, less brute-force strategy may be able to reduce their size even further. We would not be surprised if most c-command dependencies turn out to belong to IO-TSL-$(3,3,3)$ or perhaps even IO-TSL-$(2,2,2)$.

3.4.4 IO-TSL as an Upper-bound for Binding

It is possible to reduce the the complexity of (local) binding to the TSL class by doing some modifications to our tier projection and/or the string representation of the reflexives. In this section, I show how this is achieved and consequently, I propose that IO-TSL is an upper-bound for binding.

Let me start with the ITSL part. The condition that makes tier projection input-sensitive is the requirement of projecting the first element of the *c-string* on the tier. We can eliminate this requirement in various ways.

One such way is to use a *c-string* for any node that contains a reflexive. We have seen that the *c-strings* are obtained recursively, and so, if a *c-string* contains a reflexive, it will contain the *c-strings* for that reflexive as well. Using such a *c-string*, we can modify our tier projection to project the reflexive irrespective of where it appears in the input string. The rest of the tier projection process stays the same, but this eliminates the input-sensitive part, and reduces the complexity from IO-TSL to OTSL.

Consideting the OTSL part, the condition that pushes the binding tier projection from TSL to OTSL is the ϕ-compatibility condition on the potential antecedent D head that needs to be projected on the tier. If we can find a way that eliminates the need for such condition, we can drop the output-sensitive requirement in the IO-TSL tier projection proposed. If we separate the problem of finding an antecedent from a ϕ-compatible antecedent, we can avoid the output-sensitive tier projection. This would mean that our tier projection would project the reflexive, any D head and any binding domain head, call it H. The forbidden factor then would be {RH}. This would reduce the complexity from IO-TSL to ITSL. Combined with what I proposed int he previous paragpraph, one can argue that local reflexive binding fits into the class of TSL languages. However, this would mean that local binding is actually a blocking effect, with some D showing up between R and H on the tier to avoid the constraint violation.

Another way to eliminate the input-sensitive tier projection is by having a mapping from the *c-string* of an LI to a *c-string* that excludes that LI. To elaborate, we can map the "*cs*(himself)= himself John likes T C" to "John likes T C". We can use the same tier projection mechanism that projects evey D head and a binding domain head, here T. The requirement then would be that the tier should not start with a T, or in general, with a head of a binding domain, H.

In similar ways, it can be argued that IO-TSL is an upper-bound for non-local binding as well. If we follow the same mapping from the *c-string* of an LI to a *c-string* that excludes that LI, non-local binding can be reduced to OTSL. This is due to the fact that we still require to have the ability to only care about the DPs after the leftmost T in the *c-string*, since such DPs can license binding. In general, we know that SE reflexives do not have ϕ-features. This itself removes the need to look for a ϕ-compatible D head in the tier projection and the output-sensitive tier projection is avoided. Following the mechanism I suggested above about using any *c-string* that contains a reflexive, non-local binding's complexity can also be reduced to TSL.

3.4.5 Limitations of IO-TSL

IO-TSL makes some strong predictions about what shape linguistic dependencies can take and how multiple c-command conditions may be interwoven.

Example 18. Consider an unattested variant of the long-distance Principle A that applies to Swedish *sig*. In this variant, the reflexive and the antecedent not only have to be separated by a T-head, but each intervening T-head must be c-commanded by some functional head F that is lower than the next higher T-head. So the c-string language is not just $\cdots D[\phi] \cdots T \cdots R[\phi]$, but rather $\alpha R[\phi]$ such that I) α contains at least one $D[\phi]$ and, II) F occurs between every two instances of T.

This string language is not IO-TSL because we can no longer omit projecting all T-heads to the tier. To see this, contrast the well-formed pattern $D[\phi](F\ T)^*\ F\ T\ (F\ T)^* R[\phi]$ against the ill-formed $D[\phi](F\ T)^*\ T\ (F\ T)^*R[\phi]$. The only difference is the unlicensed T-head in the middle, so this T-head must end up on the tier in order to distinguish well-formed from ill-formed strings. But there is no context that can uniquely identify just this T-head without projecting other T-heads or F-heads. The set of well-formed tiers then would be an infinite subset of $\{T,F\}^* D[\phi]\,\{T,F\}^* R[\phi]$, which is not SL unless the distance between $D[\phi]$ and $R[\phi]$ is finitely bounded. Since there are no additional factors that guarantee such a bound on the distance between reflexive and licensor, our unattested variant of *sig*-licensing is not IO-TSL.

This example highlights a crucial limitation of IO-TSL: a long-distance dependency cannot apply across an unbounded number of long-distance dependencies that interact with it. The F-licensing of T-heads would be unproblematic if it were an independent constraint that must always be satisfied, rather than just being an extra condition on *sig*-licensing. For then it would be a condition on all *c-strings* that start with T-heads and could be omitted in *c-strings* that start with reflexives. Alternatively, F-licensing of T-heads could be captured if the distance between F and T were locally bounded. In this case, one could project F and T only once and skip all other locally licensed T-heads. Similarly, it would suffice to project at most one unlicensed T-head. The result would be tiers of the form $D[\phi]\alpha R[\phi]$ where α is $T\ F\ T$ or $F\ T\ T$. Then the tier language would once again be finite. So a long-distance dependency can interact with other dependencies, but either the number of those dependencies or their locus of application must be finitely bounded.

It is worth mentioning that this unattested example pattern is easily defined in first-order logic, which so far has been the only safe upper bound on syntactic dependencies (Graf, 2012a). The IO-TSL perspective of c-command dependencies thus improves on previous work by ruling out some linguistically undesirable patterns.

Now that we have got the basic understanding of the complexity of binding without taking movement into account, it is time we had a look at how binding and movement interact with each other and how movement can feed or bleed binding.

3.5 Adding Movement

The discussion so far has largely ignored the effects of movement on c-command relations. However, it is well-known that movement affects binding relations. The two major movement types, i.e. A-movement and A'-movement, play a major role in how binding relations come to existence. Take a look at an example in (76a) below, where the reflexive *himself* can ambiguously refer to either *John* or *Fred*. A similar observation can be made with three possible antecedents in (76b).

(76) a. John_i wondered which picture of $\text{himself}_{i/k}$ Fred_k liked.

(Hornstein et al., 2005: 250)

b. John_i wondered which pictures of $\text{himself}_{i/j/k}$ Bill_j claimed Paul_k had bought.

(Hicks, 2009: 158)

Under the standard binding analysis, in these examples, only *John* c-commands *himself* from its surface position, and is local to it and can bind it. These cases have been analyzed by positing a derivational analysis of binding principles, which was originally proposed in Belletti and Rizzi (1988) and states that the reflexives can be bound at any time during the derivation as (77) specifies.

(77) a. NP-moved elements may be bound from the surface position or any of the trace sites (both quantificational and anaphoric binding)

b. NP-moved elements may bind from the surface position or any of the trace sites (both quantificational and anaphoric binding) (Lebeaux, 1991, ex.71)

Following this logic, we can explain the availability of different readings due to the availability of binding during the derivation as the *wh*-phrase undergoes cyclic movement, as (78) shows for (76a). The reflexive *himself* is bound by *Fred* in its base position, and after moving to its final position, it is bound by *John*.

(78) $[_{\text{TP}}$ John wondered $[_{\text{CP}}$ [which picture of himself] $[_{\text{TP}}$ Fred liked [which picture of himself]]]]

However, this approach is challenged by examples such as (79) below, which following the derivational approach should be ambiguous between *each other* referring to the embedded or matrix subject. However, this sentence only has one meaning, where *each other* is bound by the embedded subject, i.e. *the teachers*.

(79) The students asked what attitudes about each other the teachers had.

(Hornstein et al., 2005: 251)

A better strategy to account for these ambiguities is to use the **Copy Theory of Movement**, according to which a moved element leaves a copy behind, which gets deleted in the phonological component. This theory of movement indicates that the operation *Move* is a combination of the operations *Copy* and *Merge* (Hornstein et al., 2005). Having this in mind, we can analyze our example in (76a) in the following way. The *wh*-phrase that starts as an argument of *like*, undergoes movement and leaves a copy behind as (78) above shows. This derivation has two copies of the reflexive *himself* with two different binding domains, and depending on which copy is chosen, the antecedent would differ and the meaning would differ as well.

However, the reflexive cannot refer to both of the antecedents at the same time, and therefore, one has to be deleted for us to be able to establish an operator-variable relation. The reading in which the reflexive is bound by the matrix subject can be obtained by deleting the lower copy of the *wh*-phrase. Assuming that *wh*-traces are variables and the *wh*-phrases in [Spec, CP] are quantificational operators, this derivation is well-formed as the operator-variable relation is also established correctly.

(80) [$_{\text{TP}}$ John wondered [$_{\text{CP}}$ [which picture of himself] [$_{\text{TP}}$ Fred liked [~~which picture of himself~~]]]]

Establishing the second reading, where the reflexive refers to the embedded subject, i.e. *Fred* is a bit less straightforward. We cannot simply delete the higher copy, since it would result in absence of the operator-variable relation. Given that moving parts of *wh*-phrases is available cross-linguistically, one can only leave the *wh*-operator at the higher copy and delete only the *wh*-word itself in the lower copy, as (81) shows. Using this strategy, we are able to establish the operator-variable relation as well as the binding relation in the lower clause.

(81) [$_{\text{TP}}$ John wondered [$_{\text{CP}}$ [which ~~picture of himself~~] [$_{\text{TP}}$ Fred liked [~~which~~ picture of himself]]]]

The examples up to now show how cyclic movement can result in ambiguous binding relations. More examples show that reflexives remain acceptable even if their binder has moved out of the local domain. The example in (82a) shows that the copy of the antecdent within the embedded clause can act as the binder. The same holds for *wh*-movement of subjects, as (82b) shows.

(82) a. John is believed to contradict himself.

b. Which guy do you think would contradict himself in such a blatant way?

(Büring, 2005: 245-246)

In both these examples, the final landing site of the antecedent is beyond the binding domain for the reflexive. Consequently, the antecedent cannot bind the reflexive from the final landing site. Following the derivational binding and copy theory of movement, we can conclude that the antecedent binds the reflexive in the embedded (and base) position as the following derivations show.

(83) a. [$_{\text{TP}}$ ~~John~~ is believed [$_{\text{TP}}$ **John** to contradict himself]].

b. [$_{\text{CP}}$ ~~Which guy~~ [$_{\text{TP}}$ do you think [$_{\text{TP}}$ **which guy** would contradict himself in such a blatant way]]]?

Up to now, we looked at examples where the antecedent c-commands the reflexive throughout the derivation, and it is only a matter of determining at which point during the derivation, or via using which copy, the binding occurs or can occur. The second group of examples comes from the A-movement cases, where the binding relation can only be established when the binder moves over the reflexive, as (84) shows:

(84) John seems to himself [to be a nice guy]. (Lebeaux, 1991, ex.63)

In this subject raising construction, the matrix subject *John* starts off as the subject of the embedded clause [John to be a nice guy], and then undergoes A-movement to its final landing site. The subject in the embedded clause is unable to bind the reflexive, since it does not c-command it. The binding happens once the subject has reached its final destination, from where it c-commands and binds the reflexive. Ignoring the details[12], in these cases, the higher copy survives the movement and the lower copy gets deleted, (85).

(85) John seems to himself [$_{\text{TP}}$ ~~John~~ to be a nice guy].

Adding more to the picture, we can have a combination of raising over the reflexive followed by subsequent movements and yet it can bind the reflexive, (86). In this example, it's not the final copy, but the intermediate one, that does the binding.

[12]If both copies survive, this construction is in violation of Principle C of binding, which requires R-expressions to be free. This is because the lower copy of *John* would be c-commanded by *himself* in violation of Condition C.

(86) John is likely to seem to himself to be a nice guy.

(87) [$_{\text{TP}}$ ~~John~~ is likely [$_{\text{TP}}$ **John** to seem to himself [$_{\text{TP}}$ ~~John~~ to be a nice guy]

3.5.1 Modeling Movement with C-strings

Addition of movement adds some complications to modeling binding relations, especially in the cases where binding is established after movement. One reason for this is that the MGs do not include implicit movements, in the sense that they show movement as a dependency between the mover and its landing site rather than explicitly moving the mover to its final position. As you recall, the movers stay in-situ while their features percolate up in the tree. This has been to my advantage up to now, since it allowed me to establish the list of c-commanders easily using *c-strings* on dependency trees. However, in cases with movement, we require to use copies of the movers in order to account for the established binding relations.

3.5.1.1 Multiple C-strings

Let us look at the movement cases one by one. For simplicity, I call the cases where a reflexive can be bound by various antecedent as *ambiguous* cases, and the second cases where binding is established after movement as *derived binding*. In both cases, we can accommodate for movement by using copies of the movers.

Beginning with the *ambiguous* cases, it is important to note that having multiple potential antecedents isn't really a problem for my proposal here, because the goal isn't to find the actual binder but to regulate the distribution of a form. In other words, it only matters that a dependent element has at least one licensor in its *c-string*. Determining the specific reading and the specific binder is beyond the limits of the theory I am adopting here. That being said, we can look at the *c-strings* of *himself* for the example in (76a) above, repeated in (88), once in its base position and once after its movement. Following the copy strategy laid out in the previous section, such *c-strings* would be as listed below. Both these *c-strings* are well-formed as they do not involve any forbidden factor listed in Table 3.1.

(88) John$_i$ wondered which picture of himself$_{i/k}$ Fred$_k$ liked.

(Hornstein et al., 2005: 250)

(89) a. **Base Position:** *cs*(himself) = himself of pic Fred liked T which John wondered

b. **Moved Position:** *cs*(himself) = himself of pic which John wondered

We can say that once one instance of binding is established, the string is well-formed and the sentence is ruled in. That being said, the point of using *c-strings* is to identify whether a string is well-formed; it is not, by any means, to predict ambiguities. In other words, only looking at the *c-string* for *himself* in its base position seems to be enough to rule in this sentence, without the prediction of ambiguity.

This is not the full story, though. Sometimes, a dependent element is licensed only because it has moved out of a locality domain into a higher position where it is accessible to its licensor. Let us look at a minimally different sentence below, where the only possible reading is the binding between the matrix clause and the reflexive.

(90) [$_{\text{TP}}$ John wondered [$_{\text{CP}}$ [which picture of himself] [$_{\text{TP}}$ Mary liked [~~which picture of himself~~]]]]

In this case, only the higher copy of the *wh*-movement is a legitimate candidate to establish binding relations. In other words, looking at the *c-string* of *himself* in its base position is not enough to judge this sentence as acceptable. Therefore, we need a mechanism to enable MGs to allow for copies of movers to be available in the derivation. Another strategy would be to allow *c-strings* to insert such copies.

In order to separate the mover in its base position or intermediary landing site from the mover in its final position, I am going to use f_{P} and f_{L} notation (inspired by the notation used in Pasternak and Graf (2021)) to differentiate between a PF-mover and an LF-mover. I am using the LF-movement in MG's original sense to refer to the base position of the mover as in Stabler (1997), and PF-movement to refer to the final landing site. The f_{P} would indicate the final or the surface copy of a mover, while the f_{L} would refer to the base position and all other intermediary landing positions that the mover needs to cyclically go through[13]. To put it in simpler terms, the LF-movement for me refers to reconstruction, while the PF-movement refers to the mover's final and pronounced copy. That being said, the *c-strings* in (89) can be refined more to include this distinction.

(91) a. **Base Position:** *cs*(himself) = himself of pic Fred liked T which$_{\text{L}}$ John wondered

b. **Moved Position:** *cs*(himself) = himself of pic which$_{\text{P}}$ John wondered

[13]Note that Pasternak and Graf (2021) use f_{P} and f_{L} to indicate the movement types.

Once the *c-strings* are constructed, we can use the same IO-TSL mechanism to determine the well-formedness or validity of the sentence. More specifically, the added complication is determining the mover type and where in the *c-srtings* each movement type can occur. The rest of the process is, by now, the familiar process of projecting relevant elements on tiers and determining whether or not the tiers are well-formed.

I proposed using PF and LF copies as well as using copy theory of movement to address the *ambiguous* cases. Let us look at the second cases, or what I called *derived binding*. Similar to the previous case, we can accommodate for movement effects by modifying the construction of *c-strings* such that a mover is also copied into its landing sites. The *c-strings* for *himself* are as follows for the example in (84), repeated here in (92). The dependency between the reflexive and the binder cannot be established in (93a) as the binder is not in the *c-strings*.

(92) John seems to himself [to be a nice guy]. (Lebeaux, 1991, ex.63)

(93) a. **Base Position:** *cs*(himself) = *himself to to seems T C

b. **Moved Position:** *cs*(himself) = himself to to seems John

The *c-string* in (93a) is ill-formed as there is no antecedent for the reflexive to be bound by, and it violates the absolute condition, i.e. *cs(R)= $R_{[\phi]}$... . This *c-string* reflects the c-command relation for *himself* where *John* is at its base position. The (93b) is a well-formed *c-string* since the reflexive is able to find its antecedent and be bound by it. This *c-string* lists the c-commanders of *himself* when *John* is its final landing site.

In the previous case, we saw that some sort of mechanism should be added to MGs or to *c-strings* to allow for the presence of copies. This is crucial in accounting for *derived binding* cases. Otherwise, without the higher copy being visible, the *c-strings* would rule out grammatical sentences like (84). While this complicates obtaining the *c-strings* to an extent, it does not change the generalizations in this chapter so far in terms of using *c-strings* to model binding. If we follow the PF versus LF copies, the well-formed *c-string* for the reflexive in (84) will be as shows in (94).

(94) **Moved Position:** *cs*(himself) = himself to to seems John_{P}

One can say that using the PF-movement type is enough to account for this type of constructions. However, we saw in various examples, such as (82) and (86), that the binder does not or cannot bind in its final landing site. Regardless, this is similar to the *ambiguous* cases we discussed earlier, in that we can use the same strategy and look at all the possible *c-strings* for the reflexive and as long as one is well-formed, we rule the sentence as licit. For instance, we have the following three *c-strings* for the sentence in (87), repeated here in (95).

(95) [$_{\text{TP}}$ ~~John~~ is likely [$_{\text{TP}}$ **John** to seem to himself [$_{\text{TP}}$ ~~John~~ to be a nice guy]

(96) a. **Base Position:** *cs*(himself) = *himself to to seem T C
b. **Intermediate Position:** *cs*(himself) = himself to to seem John$_{\text{L}}$
c. **Final Position:** *cs*(himself) = *himself to to seem T likely is John$_{\text{P}}$

As you can see, the only well-formed *c-string* is the one in (96b), which models the intermediate landing site for the antecedent. The *c-string* for base position is ill-formed for it lacks a potential antecedent, and the *c-string* for the final landing site is also ill-formed for there is a T intervening between the antecedent and the reflexive, i.e. the antecedent is beyond the domain in which it can bind the reflexive.

Once again, the complexity here stems from the introduction of copies to the *c-strings*. Nonetheless, once the *c-strings* are established, the IO-TSL mechanism can be used to verify whether they are legit or illicit. The question of how the *c-strings* with such copies are constructed will remain open for now and is a subject for further and future research.

Lastly, I would like to touch upon cases where both the reflexive and the antecedent move, and the binding is established sometime during the derivation. Take a look at (97), in which the reflexive is not in the c-commanding domain of the antecedent *John* in the final positions.

(97) Which picture of himself does John seem to like the best?

The following shows the derivation before any movement occurred, which is the only time during the derivation that the binding can be established.

(98) ~~Which picture of himself~~ does ~~John~~ seem [$_{\text{TP}}$ John to like which picture of himself the best]?

The next step in the derivation looks like the following, where *John* has moved out of the binding domain and cannot bind the reflexive anymore. This is followed by the last step when the *wh*-phrase moves to its final position and again is out of reach of c-commanding domain of the antecedent *John*.

(99) a. ~~Which picture of himself~~ does John seem [$_{\text{TP}}$ ~~John~~ to like which picture of himself the best]?

b. Which picture of himself does John seem [$_{\text{TP}}$ ~~John~~ to like ~~which picture of himself~~ the best]?

The corresponding *c-strings* for all these three derivations are provided below. As you can see, only the first one, i.e. when both licensee and licensor are in their base position, is a valid *c-string*. Similar to other movement cases, if one *c-string* is well-formed during the derivation, the sentence is marked as acceptable. But, it is not obvious which copy can license the relation, and nor is it required for our model to predict that.

(100) a. **Base Position:** *cs*(himself) = himself of picture which$_{\text{L}}$ John$_{\text{L}}$ like T seem T C

b. **First Movement:** *cs*(himself) = *himself of picture which$_{\text{L}}$ like T John$_{\text{P}}$ seem T C

c. **Second Movement:** *cs*(himself) = *himself of picture which$_{\text{P}}$

This strategy shifts a lot of the burden to the correct construction of *c-strings*, which may be particularly complicated when both the licensor and the licensee move. However, we need to differentiate between the grammatical inferences that would produce *c-strings* from the *dependency trees* and the actual complexity of such strings. The construction of the appropriate *c-string* from a dependency tree is still definable in first-order logic and hence *subregular*, but this is a very generous upper bound, and is to be explored further. The complexity of *c-strings* themselves is still at most IO-TSL, which has been proposed as an upper bound for phonology and is also a plausible one for syntactic dependencies and in particular binding relations.

3.5.1.2 Single C-string

We can somewhat combine the multipe *c-strings* for base position and moved positions to obtain a single *c-string* for moved cases. However, the nature of the mover would dictate how this combination is done and how the domain of binding is determined.

Starting with the cases when the mover is the potential antecedent, and not the reflexive itself, we can achieve this goal by inserting a copy of the mover in each one of its landing sites. To give an example, let us look at the sentences in (84) and (82a), repeated here in (101a) and (101b), respectively.

(101) a. John seems to himself [to be a nice guy].

b. John is believed to contradict himself.

In both these examples, the mover is the antecedent. The single *c-string* for these examples, including the copies of movers in all landing sites is given below.

(102) a. *cs*(himself)= himself to to seems John T C

b. *cs*(himself)= himself John contradict to John believed is C

As you can see, the *c-string* in (102a) is a combination of the *c-strings* in (93a) and (93b). As for the *c-string* in (102b), it is also the combination of the *c-strings* for *himself* before and after the movement of *John*. In both cases, establishing one licit binding relation is enough to accept the *c-string* as well-formed.

In (102a), this relation is established between the reflexive and John once. Remember that the *c-string* of *himself* in base position was illicit. For the case in (102b), the binding is established between the reflexive and the first John that follows it. Again, this is in line with what was proposed for multiple *c-strings* cases. In this example, John can only bind the reflexive in its base position corresponding to the first copy that follows *himself* in this *c-string*. That being said, I can use this single format to account for moved antecedents.

In cases where the reflexive is the mover, the strategy is a bit different. We saw in previous section that very move of the reflexive results in a new binding relation. In the case of example in (76a), repeated here in (103), the reflexive finds antecedent in the embedded clause first (before moving), and then finds a second antecedent in the matrix clause (after moving).

(103) John$_i$ wondered which picture of himself$_{i/k}$ Fred$_k$ liked.

It is as if every movement of the reflexive establishes a new binding domain. While in the base position, the binding domain is the embedded TP, after movement, it becomes the matrix TP. In these cases, we can use the *c-string* for the base position for reflexive with the possibility that any movement of the reflexive would extend its binding domain to not be the smallest TP containing it, but the smallest TP containing its final landing site.

In this section, I showed that we can model dervied binding relations with a single *c-string*. Once we model such cases in the *c-string* format, defining the complexity is a task we should be familiar with by now. Modeling dervied binding via a single *c-string* reduces the complexity of such relations to be in parallel with base c-command cases I have extensively looked at throughout this chapter.

3.6 Conclusion

I have defined a string-based representation format over dependency trees that allows for c-command dependencies to be easily evaluated for any given node. The dependencies over these strings all fall within the class IO-TSL, which was first defined for phonology (Graf and Mayer, 2018). This chapter marks but a first step towards a *subregular* theory of syntactic dependencies, and a lot remains to be done.

The current approach only measures the complexity of a syntactic dependency with respect to a specific node. To check the whole dependency tree, one has to evaluate the c-string of each node. We do not know whether the TSL-approach of Graf (2018b) provides a method for doing so. As long as all dependencies are IO-TSL, though, the well-formedness of the whole dependency tree can be verified by a deterministic top-down tree automaton with a look-ahead of 1. This implies *subregular* complexity and may even allow for highly efficient parsing algorithms.

We saw that adding movement to the picture and the notion of *derived* c-command complicates the construction of *c-strings* as it requires adding a new mechanism to be able to include copies of movers in the *c-strings* as well (cf. Vu (2020)).

Dependencies that go beyond c-command cannot be handled with this approach. This includes binding via sub-command, parasitic gaps, and across-the-board movement as an exception to the coordinate structure constraint. It remains to be seen whether they can be accommodated with tree tiers as proposed in Graf (2018b), or whether a completely new perspective is needed for these phenomena.

The next chapter addresses yet another syntactic dependency, i.e. island constraints, and shows that these constraints are also subregular. Moreover, we will see that if a similar string-based representation is used, these constraints belong to the IBSP class, whose generative capacity is similar to TSL, even though these two classes are not directly comparable.

Chapter 4

Domain-based Non-Locality of Islands

In Chapter 3, we deal with one form of syntactic long-distance dependencies, i.e. binding conditions. The notion of *c-strings* is defined and is implemented to represent binding relations. It is shown that using this form of representation, the complexity of these binding conditions can be reduced to the realm of *subregular* string languages, which puts them in parallel with phonological constraints. This chapter addresses another syntactic phenomenon, namely *island constraints*.

The notion of syntactic islands has received great attention from the linguistic community. An island is a constituent from which no subpart may be extracted. Despite decades of research, it is still not clear why islands exist and how uniform a class they form. This chapter provides a formal framework for analyzing island constraints from a *subregular* perspective. Similar to the string representation used in the previous chapter for binding conditions, in this chapter, too, key aspects of the syntactic representation are encoded as strings. However, the difference is that in this chapter, these string representations encode dominance, rather than c-command. Island effects then are expressed as constraints on the shape of these strings. The constraints turn out to fit in the class IBSP (Interval-Based Strictly Piecewise), which has been previously explored in *subregular* phonology. Consequently, the characterization of islands in terms of IBSP string constraints not only provides a computational upper bound on the inventory of feasible island effects, but also establishes a surprising link between syntax on the one hand and phonology on the other. Another upshot of this analysis is unifying island constraints under one natural class.

4.1 Background

The idea of syntactic islands originates from Chomsky's Transformational Grammar, where it was first discussed by Ross (1967). Ross (1967) noticed that movement dependencies are illicit in some structures, generally known as **islands**. For instance, consider the case of English *wh*-phrases. While the movement of the *wh*-phrase *what* in (1) is licit, the same movement is ill-formed in the following examples in (2), which are what Szabolcsi (2006) calls classical strong islands (examples (a) to (d) are taken from Boeckx (2012), and example (e) is from Szabolcsi (2006)). This is because a syntactic dependency between a gap and its antecedent cannot be established inside an island (Szabolcsi, 2006).

(1) What did Bill buy _?

(2) a. * What did you hear rumors [that John bought _]? *Complex NP*
b. * What did you eat [ham and _]? *Coordinate Structure*
c. * Who did [that Mary kissed _] bother you? *Sentential Subject*
d. * Whose did you buy [_ book]? *Left Branch*
e. * Which topic did you leave [because Mary talked about _]? *Adjunct Island*

The *wh*-movement out of the aforementioned constructions always results in ungrammaticality regardless of whether the mover is an argument or an adjunct, hence the term **strong islands**. However, not always do we have the same degree of ungrammaticality caused by movement. The structures in which the movement of an argument is slightly better than the movement of an adjunct are called **weak islands**, example of which is *wh*-islands, as shown in (3).

(3) a. ?? Who/Which girl do you wonder [whether Bill kissed −]?
b. * How/In what way do you wonder [whether Bill kissed Sue −]?
(Boeckx, 2012: 17)

Apart from the nature of island constraints, some of these constrains show cross-linguistic variation. For instance, while the *left branch* condition, shown in (2d), penalizes the movement of an NP on the left branch of a bigger NP in English, it does not hold for some other languages such as French (Sabel, 2002), as (4) shows.

(4) a. Combien de problémes sais-tu résoudre −? **French**
How many problems can you solve?

b. Combien sais-tu résoudre [− de problémes]? (Sabel, 2002: 288)

Other cross-linguistic differences are extraction out of subjects, which is ungrammatical in English but fine in languages such as Japanese, Navajo, Turkish, Palauan, Hungarian, and Russian (Stepanov, 2007). The next two sections briefly discuss these two accounts, namely Sabel (2002) and Stepanov (2007), for being amongst the well-received analyses in the Minimalist framework and for their attempts at unifying island constraints.

4.1.1 Sabel 2002

As mentioned at the begging of this section, Ross (1967) was first to examine these constraints. A great number of scholars, such as Chomsky (1986a), Rizzi (1990), Cinque (1990), Takahashi (1994), Postal (1998), Nunes and Uriagereka (2000), Starke (2001) among others, have since made attempts to come up with analyses that can account for the variety of these effects within different frameworks. These analyses all belong to the Government and Binding era, whose notions are not in use anymore. Here I focus on the research made after Chomsky (1995)'s Minimalist Program came into existence, in an effort to give a succinct overview of some of the existing accounts. Perhaps, the first such proposal is Sabel (2002).

In an attempt to provide a unified account for both strong islands and *wh*-islands, Sabel (2002), adopts the proposal by Chomsky and Lasnik (1993) that a trace is *-marked if its generation involves a violation of a locality constraint, namely a violation of a *wh*-island or a strong-island (or CED-island[1]) constraint. This *-feature is brought into the derivation by the *wh*-phrase, which becomes visible if the *wh*-phrase violates a movement condition on its movement path. The mild violation cases of weak islands (*wh*-islands) are due to

[1]Beginning with Huang (1998)'s Condition on Extraction Domains (CED), provided in (i.), the ungrammaticality of movement out of adjuncts and/or subjects is linked to proper government or lack thereof.

i. CED: A phrase A may be extracted out of a domain B only if B is properly governed.

Proper government is defined under the Empty Category Principle (ECP), which requires a nonpronominal empty category to be either (i) θ-governed or (ii) antecedent-governed (Rizzi, 1990).

What this means is that the moved element should either be θ-marked by a lexical head, e.g. objects and manner adverbs are θ-marked by the verb, or it has to move to a position where the moved elements itself and its trace can form an uninterrupted link.

deletable *-features, while in the cases of strong islands the *-features are not deletable and therefore, remain visible at LF causing severe ungrammaticality. To elaborate a bit further, the deleted features are part of a uniform chain (5), while the undeletable ones are extracted from inside a (modified) definition of a *barrier* (6).

(5) **Uniformity Condition on Chains (UCC)**

a. A chain C is uniform with respect to P (UN[P]) if each α_i has property P or each α_i has property non-P.

b. i. A' ... (A') ... A (Operator-variable construction)

ii. θ' ... (θ') ... θ (Sabel, 2002: 281)

(6) **Barrier**: A category A may not be extracted from a subtree T_2 (X^{max}) of T_1 if T_2 was merged at some stage of the derivation with a complex category (i.e. with a non-head). (Sabel, 2002: 292)

Giving this UCC condition, if a *wh*-movement path forms a uniform chain, none of the intermediate traces can be deleted. Therefore, if one of these traces violates a movement condition, the sentence would be completely ungrammatical. For instance, an adjunct *wh*-phrase starts in a non-θ-position and moves to a non-θ-position and forms a uniform chain; therefore, its intermediate trace cannot be deleted in the following example:

(7) *[$_{CP}$ How$_2$ [$_{VP}$ do you wonder [$_{CP}$ *t_2' what$_1$ [$_{C'}$ [$C_{[wh1][wh2]}$] [John could fix t_1 t_2]]]]]?

Comparing this sentence with the less ungrammatical sentence in (8), we can see that the *wh*-movement path in this example is not uniform because the base position of the *wh*-phrase is a θ-position when its landing site is not. As a result, the intermediate $*t$ is deleted and hence, the mild violation.

(8) ??[$_{CP}$ What$_2$ [$_{VP}$ do you wonder [$_{CP}$ *$\cancel{t_2'}$ how$_1$ [$_{C'}$ [$C_{[wh1][wh2]}$] [John could fix t_1 t_2]]]]]?

It should be noted, as the reader might have noticed, that Sabel (2002) uses multiple CP specs with multiple *wh*-features which is necessary for his analysis to be able to tease apart the differences between *wh*-islands and other movement violations.

Moreover, to extend this analysis to strong islands, Sabel (2002) employs a modified definition of Chomsky (1986a)'s notion of barriers to prevent extraction out of any phrase that is not merged with a head, as given in (6) above. The reason for this need is because utilizing the UCC alone is not sufficientt to account for the severity and consistency of ungrammaticality in these sentences. What this gains is that it prohibits the extraction out of a subject (9a), an adjunct (9b), and a complex NP assuming that the CP is not the argument of the noun phrase (9c) (Sabel, 2002).

(9) a. *[$_{CP}$ How did you [$_{VP}$ [$_{VP}$ leave] [$_{PP}$ before [$_{CP}$ *t' solving the problem t]]]]?

b. *[$_{CP}$ How did you hear [$_{NP}$ a rumor [$_{CP}$ *t' that John had solved the problem t]]]?

c. *[$_{CP}$ How would [$_{CP}$ *t' for John to fix the car t] be difficult]?

(Sabel, 2002: 292)

Generally, the modified definition of a barrier excludes extraction out of phrases that are merged with non-heads. Basically, almost anything except for objects, is merged with a phrase or a bar-level projection. For instance, subjects are merged with a VP (or a *v*P), and so extraction out of them is ill-formed. In all the examples above extraction is illicit for the very same reason. In (9a), the subject CP has merged with a v' or a T$'$; in (9b), the adjunct PP is a barrier because it was merged with VP at some stage of the derivation; and in (9c), the adjunct CP is a barrier for t because the CP was merged with DP at some stage of the derivation.

While Sabel (2002)'s analysis looks very appealing, it is at best stipulation. It is not clear at all why the intermediate traces induce violations. They induce violations because they move out of islands, but this is exactly what we need to capture, not the other way round.

The even bigger issue it runs into is unifying subjects with adjuncts, which as Stepanov (2007) points out, do not form a natural class. While extraction out of adjuncts is cross-linguistically bad, there are languages, such as Japanese, Navajo, Turkish, Palauan, Hungarian, and Russian (Stepanov, 2007), that do allow for extraction out of subjects, as the following examples show for Turkish and Russian, respectively.

(10) a. [Op_i [*pro* [t_i anne-si]-nin herkes-le konuş-tu-ğ-u]-nu
[Op_i [*pro* [t_i mother-AGR]-GEN everyone-with talk-PST-COMP-AGR]-ACC
duy-du-ğ-um] adam.
hear-PST-COMP-AGR] man
'(lit.) The man [whose I heard [that [_ mother] talked to everyone]].'

b. S kem by ty xotel čtoby govorit' bylo by odno
with whom SUBJ you wanted that-SUBJ to-speak were SUBJ one
udovol'stvie?
pleasure
'(lit.) with whom would you want that [to speak _] were sheer pleasure?

(Stepanov, 2007: 90-91)

For this very reason, Stepanov (2007) suggests an "eclectic" approach that classiffes subjects and adjuncts under two different categories. I will briefly discuss his analysis in the next section.

4.1.2 Stepanov 2007

In his analysis of islands, Stepanov (2007) utilizes the proposals made in Takahashi (1994) and Nunes and Uriagereka (2000), arguing that extractability out of subjects should be explained under Takahashi (1994)'s "Chain Uniformity" approach, while the uniform lack of extractability out of adjuncts needs to be explained via Nunes and Uriagereka (2000)'s "Structure-Building" approach.

Following this logic, Stepanov (2007) provides a minimalist analysis in which the possibility of extraction out of subjects is tied to their failure of movement to [Spec, TP]. Stepanov (2007) generalizes Takahashi (1994)'s proposal for Japanese explaining that in morphologically rich languages with an explicit tense morpheme, the subjects remain in VP. The process of *wh*-extraction out of such subjects goes through the intermediary position of [Spec, TP] followed by an English-like *wh*-movement to [Spec, CP]. These two steps are required because of two general constraint proposed in Takahashi (1994), namely Uniformity Corollary on Adjunction (UCA), and Shortest Move, as defined in (11) and (12), respectively.

(11) **Uniformity Corollary on Adjunction (UCA):** Adjunction to a part of a nontrivial chain or coordination is not allowed. (Stepanov, 2007, ex.7)

(12) **Shortest Move:** Make the shortest move. (Stepanov, 2007, ex.6)

The Shortest Move constraint enforces movement to proceed as a series of successive adjunctions of the moving element to the maximal categories along its path. Take the following example, for instance:

(13) ?*Who$_i$ does [a picture of t_i] hang on the wall? (Stepanov, 2007, ex.1)

In this sentence, the [$_{DP}$ a picture of who] raises from [Spec,*v*P] to [Spec,TP], forming a two-linked chain ([$_{DP}$ a picture of who], [$_{DP}$ a picture of who]). This is followed by the extraction of *who* from the higher link and its adjunction to this DP, making a shortest move. Now, we have the chain: [$_{DP}$ who [$_{DP}$ a picture of who], [$_{DP}$ a picture of who]]. Since this chain is not uniform, due to its links having different structures, the sentence is ruled out.

In Japanese-type languages, it is argued that the subjects stay within the VP. As a result, the Shortest Move enforces the *wh*-operator movement to go through adjunction to the clausal subject itself. Since the unmoved subject is not a non-trivial chain (being a single-membered chain), this movement crucially does not violate the UCA. Subsequently, the clausal subject adjoins to higher projections until it reaches its destination, the matrix [Spec, CP]. As a result, extraction out of subject is these languages does not induce ungrammaticality.

We can, in general, adopt the following constraint from Diesing (1992) to rule out extraction out of English-type subjects[2].

(14) **Diesing's Extraction Constraint:** Extraction cannot take place out of an NP that must raise out of VP. (Stepanov, 2007, ex.47)

As mentioned above, Stepanov (2007)'s objective, dictated by the cross-linguistic variations in terms of extraction out of non-complements, is to provide separate analyses for subjects and adjuncts. In regards with the extraction out of adjuncts, which he reports are universally opaque, his analysis relies on the Late Adjunction Hypothesis (LAH), in (15).

(15) **Late Adjunction Hypothesis**: Any adjunction must take place after all instances of substitution Merge have applied (in other words, postcyclically).

(Stepanov, 2007: 112)

Take an example like (16) below.

(16) John went to bed [$_{Adjunct}$ after Peter fixed what]

[2] Another similar proposal can be found in Rizzi (2007) and Rizzi and Shlonsky (2007) on criterial freezing, which requires a Criterial Goal to be frozen in place. In other words, a phrase that has moved cannot move again. Please refer to the aforementioned sources for more details.

As Stepanov (2007) explains, the adjunctive phrase *after Peter fixed what* is a thematic and structural adjunct with no uninterpretable features. Therefore, by LAH, it has to be Merged postcyclically. This sentence has two phrases as shown below:

(17) a. $[_{\text{CP}}$ $[_{\text{IP}}$ John went to bed]]

b. $[_{\text{Adjunct}}$ after Peter fixed what] (Stepanov, 2007, ex.62)

Since the adjunct here is merged after the CP is built, it is not merged by the time the interrogative feature Q of the matrix complementizer is merged with the IP, *John go to bed*. Therefore, this Q feature remains unchecked (it must be checked at the point of insertion), and the derivation crashes.

While Stepanov (2007) seems to be on the right track in differentiating between various non-complement types rather than trying to unify them under the same mechanism, as Boeckx (2012) points out, the evidence offered to justify a heterogeneous treatment of the CED is rather weak. To begin with, Boeckx (2012) mentions that the cross-linguistic differences in respect with extraction out of adjuncts versus extraction out of subjects could be due to reasons other than islandhood. For instance, what counts as a subject in various languages might differ from each other, and therefore, when dealing with extraction out of subjects, we might be dealing with different domains. Secondly, it is not clear how Stepanov (2007)'s LAH analysis can account for, what are known as, Truswell (2007) sentences, as in (18) below, in which extraction out of adjuncts is perfectly fine:

(18) a. $[_{\text{CP}}$ What$_i$ did you come round [in order to work on t_i]]?

b. $[_{\text{CP}}$ What$_i$ did John arrive [whistling t_i]]? (Truswell, 2007: 117-18)

As this section has shown, island effects are not as clear-cut as one might hope for, and they are poorly understood even empirically. While linguistic research has drawn up a fine-grained typology of islands, theoretical linguistics still lacks a strong, unified theory of islands. The main conceit of this chapter is that a computational perspective rooted in *subregular* complexity provides a rough yet insightful approximation of the limits of this class, by unifying them under a single class while covering cross-linguistic variation. Using a similar string representation to what we saw in Chapter 3, and following the intuition in the next section, I show that islands form a natural class computationally, in that they all fall under the *subregular* class of IBSP languages.

I focus mainly on strong islands, with a brief note on how the proposed class can be extended to weak islands as well. I start with the least controversial island constraints, followed by a discussion of less-understood exceptional cases as well as cross-linguistic variation. I will also attend to some other constraints on movement, such as the *that*-trace effect at later points.

The chapter proceeds as follows: §4.2 provides an intuitive view on island constraints. This is followed by an introduction of the IBSP language classes in §4.3. We then use Minimalist grammars, discussed in §2.1, to define a string-based encoding of dominance relations, which leads to our central result in §4.4 that a number of island constraints are at most IBSP-2 (and can even be further reduced to IBSP-1). I conclude with a discussion of the formal and linguistic implications of this finding (§4.5).

4.2 Islands: An Intuitive Approach

Remember from §2.1 that movement in MGs is a form of dependency between the mover with a *licensee* feature and the landing site with a matching *lisencor* feature.

As mentioned earlier, islands are constituents, movement out of which generally results in ungrammaticality. In other words, they prevent the dependency from being established between a mover and its landing site (between the probe and the goal). Let us have a look at our example (1), repeated here in (19a), and compare it to the ill-formed sentence in (19b), which involves movement out of an adjunct.

(19) a. What did Bill buy _?
b. * What did you laugh because Bill bought _?

Let us represent these sentences using a copy theory of movement, and replace the gaps with their deleted copies (ignoring the intermediate copies). Let us also use bracketing to show categories of phrases in the sentences.

(20) a. [$_{\text{CP}}$ What [$_{\text{TP}}$ did Bill buy ~~what~~]]?
b. *[$_{\text{CP}}$ What did you laugh [$_{\text{AdjunctP}}$ because [$_{\text{TP}}$ Bill bought ~~what~~]]]?

Looking at the sentences above, we can see that we have the following configuration for the well-formed sentence, where the dependency is established across a TP.

(21) what ... TP ... ~~what~~

What we can see is that, in the well-formed sentence, the relation or dependency between the two copies of the *wh*-phrases is not interrupted. This, in a sense, is similar to Sabel (2002)'s "barrier" account, which I touched upon in the previous sections. In the well-formed sentences, the *wh*-phrase can move out because it is inside a phrase that is merged with a head. Whereas, for the ill-formed sentence, the *wh*-phrase is forced to move out of a phrase that is merged with a maximal projection, i.e. the adjunct phrase merges with a TP. Intuitively, we can say that the adjunct acts like a "barrier" or, what I am going to call, a blocker. Schematically, we can show this using the following configuration. The presence of the adjunct prevents the two copies of the *wh*-phrase to talk to each other.

(22) * what ... AdjunctP ... ~~what~~

This notion of interruption should be familiar by now. We saw in the previous chapter, that the dependency between a reflexive and its potential binder should not be interrupted by some specific elements, such as a T head or a D head. A similar thing is observed here.

This intuition can be extended to other island constraints. See the following example in (23b) for a complex DP/NP constraint, compared to a well-formed sentence in (23a).

(23) a. [$_{\text{CP}}$ What did you hear [$_{\text{CP}}$ that Bill bought ~~what~~]]?
  b. *[$_{\text{CP}}$ What did you hear [$_{\text{DP}}$ the rumor [$_{\text{CP}}$ that Bill bought ~~what~~]]]?

If we represent these two sentences with similar configurations as above, we end up with the following representations for the well-formed versus the ill-formed sentence:

(24) a. what ... CP ... ~~what~~
  b. * what ... DP[CP] ...~~what~~

As these configurations show, it is not the case that any maximal projection can intervene between the two copies. For instance, for Sabel (2002), VPs and TPs are not barriers since they merge with heads T and C, respectively. Here, we see that CP is not a barrier either as it merges with V head. However, we are going to see that in the cases of the *that*-trace effect, this notion of barriers is not going to help. This is partly the reason why the *that*-trace effect does not entirely fit the class of islands in general and as defined in this chapter.

The notion of blocker introduced in this section is not new and not exclusive to syntax, either. Blocking effects have been reported in phonology too, as seen in previous chapters. Some such phenomena are non-final RHOL (Rightmost Heavy, Otherwise Leftmost) stress pattern (originally discussed in Hayes (1995), but see Baek (2017) and Graf (2018a) for computational analyses of the phenomenon), and non-local blocking of local dissimilation in Samala (original data from Applegate (1972), see McMullin (2016) and Graf (2018a) for computational discussions). To refresh the reader’s mind, I go over this blocking effect in more detail in Samala.

Samala displays a regressive sibilant harmony that requires the two sibilants in a word to agree in anteriority (irrespective of the distance between them) (McMullin, 2016) (based on Applegate (1972)). That means that a word with the underlying form /k**s**luk’ilimekeken**ʃ**/ would surface as [k**ʃ**uk’ilimekeket**ʃ**], and not [k**s**uk’ilimekeket**ʃ**]. Moreover, the language shows a local dissimilation process that turns /sn/ into [ʃn], /sl/ into [ʃl], and /st/ into [ʃt]. However, when there is a sibilant harmony, local dissimilation is blocked. While [snetus] is well-formed, [**ʃ**netu**s**] is not. [**s**netu**s**] respects the sibilant harmony at the cost of violating the dissimilation. As mentioned earlier, this blocking effect is non-local, similar to the blocking effect I sketched above for island effects.

The idea of “blocking” effect is the intuition that the rest of the chapter hinges on. In the sections to come, I discuss these island effects in an effort to unify them under a single class using this intuition. Therefore, the main effort is on identifying the class of blockers that not only can unify various islands under the same class, but can also address cross-linguistic differences. To do so, I start with English as the base language and extend the discussion to other languages, and exceptional cases. Once these blockers are identified, defining their computational complexity becomes a task of using them as forbidden factors, similar to what we did for binding effects. Before diving deeper into the world of islands, I explain the IBSP language class.

4.3 Interval-Based Strictly Piecewise (IBSP)

The *subregular* languages were discussed extensively in Chapter 2. In this section, I explain an extension of the class SP, namely the IBSP class, which uses intervals to restrain the domain that the SP grammar can apply to. Following that, I show that this class encompasses and unifies island constraints naturally. But before attending to these, I introduce a new string representation that I use to analyse island constraints, i.e. *ancestor strings*. This new representation differs from *c-strings* in that it encodes proper dominance in the dependency tree, however, the central idea of using a string representation for syntactic relations is still intact.

Interval-Based Strictly Piecewise class, originally proposed in Graf (2017) as Domain-Based Strictly Piecewise, is a means of addressing the unboundedness of the class SP. Similar to how TSL is used to localize unbounded dependencies by projecting the relevant elements on a tier, IBSP is used to bring the notion of locality in the SP class by defining intervals, within which an SP grammar can be evaluated. For a formal definition, please refer to §A.5.

The original definition of IBSP in Graf (2017), which is rooted in first-order logic, does not allow for multiple filler specifications. Hence it is too weak for some phenomena such as Korean vowel harmony. The more general version is described in Graf (2018a), but no formal definition is provided. Shafiei and Graf (2020) give a more evolved definition by making use of regular expressions.

An interval is defined via three components: I) the left and right domain edges shown with teal blocks, II) a finite number of open slots (the number matches the length of the constraint's n-grams) shown with blue blocks, and III) the fillers that may occur between the open slots shown with red blocks. This is schematically shown in Fig. 4.1 below. In this diagram, the left and right edges are showing the domain or the interval that the SP grammar applies to. The SP grammar itself is defined in the blue block, specified by *my grammar* in the diagram. Lastly, the red blocks define what other conditions need to be met or what context the SP grammar requires.

Figure 4.1: Schematic IBSP Grammar

Example 19. Unbounded Tone Plateauing is IBSP across words: We saw previously, in Example 6 in Chapter 2, that UTP is SP_3 using the forbidden factor of *HLH* to rule out ill-formed examples where a Low tone appears between two Hight tones. While this works fine for individual words, it would fail in cases where a string contains more than one word, e.g. \$LHL\$LHL\$. Since SP does not take into account word boundaries, it would incorrectly rule out this well-formed string of words. This issue is resolved using IBSP and defining the interval to be the word boundary, symbolized by $ sign, as the diagram below shows.

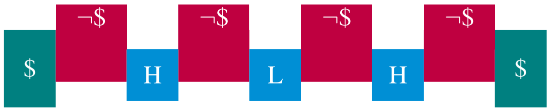


Figure 4.2: UTP across words is IBSP

This diagram shows that our SP_3 grammar of *HLH* is now only evaluated within a specific locality domain, namely the interval covering the span between two word edge markers. Within this interval, there are three open slots appearing within the word boundaries, hence ¬$ fillers. This shows that no other word edge marker occurs between the slots or the domain edges. The string is ill-formed iff at least one of those open slot configurations matches the trigram *HLH*.

Let us quickly verify how this IBSP grammar rules out $LHLLHL$ while permitting LHLLHL$. In $LHLLHL$, there are only two word edge markers, so these are the only available choices for the domain edges. We now can pick the two Hs and put them in the first and third open slot. For the second open slot, we pick one of the Ls that occur between the Hs. Since no word boundaries occur between any of those tones, this is a licit way of filling the open slots and since we have been able to instantiate the trigram *HLH* this way, we can correctly rule out the illicit $LHLLHL$ string. On the other hand, for LHLLHL$ string, there are three edge markers, so we can start by going from left to right and picking the first and second boundary markers. This would leave us with LHL string. Now, if we fill our slots with these symbols, the forbidden trigram of *HLH* is not instantiated. Therefore, the first part of the LHLLHL$ is well-formed. We can follow the similar procedure for the symbols between the second and the third boundary markers and still be unable to instantiate the forbidden SP_3 factor. As a result, the whole string is ruled in correctly.

The IBSP grammar for UTP across words can be simplified to IBSP-1 instead of IBSP-3 by using the H tone symbols as edge markers and forbidding a L tone symbol to fill the open slot. This is shown in Fig. 4.3 below.

Example 20. Korean vowel harmony is IBSP: Korean vowel harmony distinguishes three types of vowels: bright (B), mid-dark (M), and high dark (H). Within the same word, B and

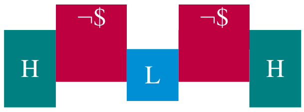


Figure 4.3: Simplified IBSP grammar for UTP across words

M cannot co-occur because of vowel harmony. The status of H is more involved. In general, H is not subject to vowel harmony. But if H is the first vowel of the word, it cannot co-occur with B. Hence there are three string patterns for sequences of Korean words (with C as a shorthand for any consonant and \$ as a word edge marker): $\$(C^{*}B\,\{B,H,C\}^{*}\,\$)^{+}$, $\$(C^{*}M\,\{M,H,C\}^{*}\,\$)^{+}$, and $\$(C^{*}H\,\{M,H,C\}^{*}\,\$)^{+}$.

Figure 4.4: IBSP locality domain for Korean vowel harmony

IBSP captures this behavior via two components, as Fig. 4.4 illustrates. First, an interval is used to define a locality domain for vowel harmony. This figure encodes that the grammar will put a constraint on two *open slots* (in blue), which are to be filled with segments. But open slots are evaluated only within a specific interval. Above, the left and right edge must be \$, i.e. word edges. There must not be any vowels between the left edge and the first open slot. That is to say, only consonants are possible *fillers* between the left edge and the first open slot. In addition, there may not be any word edges between the first and the second open slot, nor between the second open slot and the right word edge. This interval specification captures that vowel harmony cannot apply across words, and that the first vowel is special. The second component is a list of illicit combinations for the open slots: BM, MB, and HB. A string is well-formed iff there is no way to instantiate the interval in such a way that the open slots contain an illicit combination.

This IBSP grammar correctly enforces Korean vowel harmony. For instance, \$CHCMCB\$CM\$ is illicit because one can instantiate the interval with a forbidden open slot configuration. The left and right edges are lined up with the first two word edges, the first open slot is H, and the second one is B. All of the remaining material can be fillers in this interval. On the other hand, \$CHCM\$CB\$ is well-formed. Suppose that the two open slots contain H and B, respectively. Then the area between them necessarily includes

$, which is not a permitted filler in this region. None of the licit ways of instantiating the interval in $CHCM$CB$ can ever yield a forbidden combination of open slots. Note that the string CHCMCB would also be permitted as the interval cannot be instantiated at all without $ — so this specific account presupposes the presence of word edge markers in the string.

4.4 IBSP Analysis of Syntactic Islands

We saw in Chapter 3 that c-command relations can be encoded in the form of strings, using a formula to obtain them from dependency trees. A similar mechanism is used here to obtain *a-strings*, roughly replacing dominance, from dependency trees.

I could in principle use the system put forth in Graf and Shafiei (2019) to give us the *c-strings* or the list of c-commanders of a node. Since movement always happens to a containing c-commanding node, and given that *c-strings* provide a list of c-commanders both containing and non-containing, I use *a-strings* to model islands instead of c-strings. This will allow me to avoid the need to filter out non-containing c-commanders, which will be specifically relevant to the analysis I provide for amelioration effect in *that*-trace cases.[3]

The class IBSP is used to account for island effects to differentiate between *licensing* conditions and *constraining* effects. In other words, this distinction is done on the basis of the type of the dependencies, rather than their mathematical properties. That being said, island effects can still most probably be modeled in the class of TSL languages. My analysis here in the IBSP class does not rule out such possibility.

4.4.1 Obtaining Ancestor Relation as Strings (*a-strings*)

This relation is defined below in (25). Put it other way, *a-strings* (*as*) corresponds to dominance over dependency trees.

(25) *a-strings* : If a node x is part of the constituent represented by node y, we say that y contains or dominates x. In other words, (y, x) $\in$ *as*.

[3] It is worth mentioning that using *c-strings* to define the complexity of island effects gives similar results as laid out in Shafiei and Graf (2020). However, the distinction between island effects and other types of movement constraints is better explained via *a-strings*.

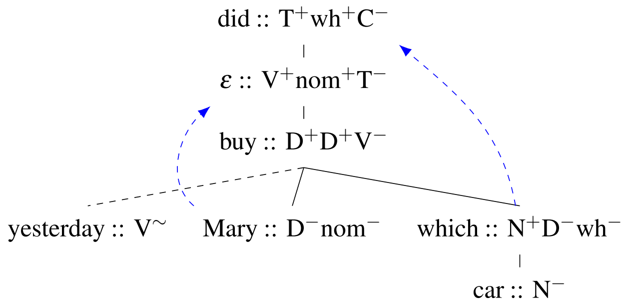


Figure 4.5: Dependency tree representation of MG derivation for *which car did Mary buy yesterday*

Let me elaborate this using the very example we used in Chapter 3 to obtain *c-strings*, repeated here in Fig. 4.5.

To obtain *c-strings*, we concluded that one needs to go left and up the dependency tree. The *a-strings* for a given node are obtainable by moving upward in the tree, just like the way we gather the dominance relation. Therefore, we have the following *a-strings* for the nodes in the tree.

- *as*(car)= car :: N^- which :: $N^+D^-wh^-$ buy :: $D^+D^+V^-$ ε :: $V^+nom^+T^-$ did :: $T^+wh^+C^-$
- *as*(which)= which :: $N^+D^-wh^-$ buy :: $D^+D^+V^-$ ε :: $V^+nom^+T^-$ did :: $T^+wh^+C^-$
- *as*(Mary)= Mary :: D^-nom^-buy :: $D^+D^+V^-\varepsilon$ :: $V^+nom^+T^-$did :: $T^+wh^+C^-$
- *as*(buy)= buy :: $D^+D^+V^-$ ε :: $V^+nom^+T^-$ did :: $T^+wh^+C^-$
- *as*(ε)= ε :: $V^+nom^+T^-$ did :: $T^+wh^+C^-$
- *as*(did)= N/A (ROOT)

We are finally in a position to show that syntactic islands not only fit into the class IBSP, but do so in a natural fashion. In the next sections, I use this machinery to analyze syntactic island effects from a *subregular* perspective. I show that islands can be likened to blockers in phonology, which interrupt the licensing relation between two elements. Island effects are established to be expressible as IBSP string constraints.

I first start with well-formed sentences using ancestor relation on the dependency trees as our representation. Then, I move to different islands and by making comparisons between their representations and those of well-formed sentences, I provide IBSP constraints that correctly separate the well-formed *a-strings* from the ill-formed ones. I first start with

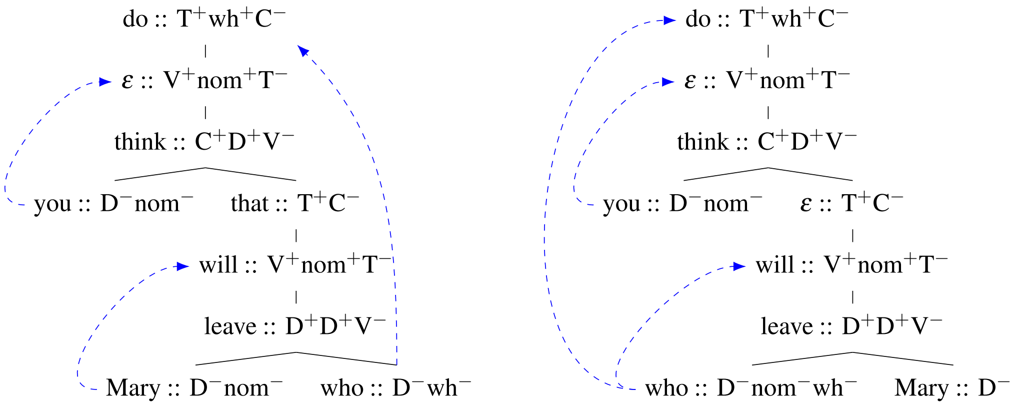


Figure 4.6: Dependency trees for *Who do you think that Mary will leave?* (left) and *Who do you think will leave Mary?* (right)

less-controversial constraints moving my way through exceptions and cross-linguistic variation. Consider first the two well-formed sentences below.

(26) a. Who do you think [that Mary will leave _]?

b. Who do you think [_ will leave Mary]?

Their corresponding dependency trees are depicted in Fig. 4.6, and the corresponding *a-strings* (*as*) for *who*, which basically is a list of the nodes dominating it, are listed in (27). Take the tree on the left in Fig. 4.6 as an example. Starting at the root node, *do* :: T$^+$wh$^-$C$^-$, we can see that it contains all the other nodes in the tree. In other words, it contains its daughter and their daughters. Moving down the tree, the second node, ε :: V$^+$wh$^+$T$^-$, contains *think* :: C$^+$D$^+$V$^-$ and all its daughters, and so on. As the reader can verify, *a-strings* are the equivalent of proper dominance.

(27) a. *as*(who) = who :: D$^-$wh$^-$ leave :: D$^+$D$^+$V$^-$ will :: V$^+$nom$^+$T$^-$ that :: T$^+$C$^-$ think :: C$^+$D$^+$V$^-$ ε :: V$^+$nom$^+$T$^-$ do :: T$^+$wh$^+$C$^-$

well-formed object wh-movement

b. *as*(who) = who :: D$^-$wh$^-$ leave :: D$^+$D$^+$V$^-$ will :: V$^+$nom$^+$T$^-$ ε :: T$^+$C$^-$ think :: C$^+$D$^+$V$^-$ ε :: V$^+$nom$^+$T$^-$ do :: T$^+$wh$^+$C$^-$

well-formed subject wh-movement

As the MG features make the list very verbose, I use a more compact notation that only keeps track of the features that will matter for our analysis.

(28) a. *as*(who) = who[wh$^-$] leave will that[C$^-$] think T C[wh$^+$]

well-formed object wh-movement (simplified)

b. *as*(who) = who[wh$^-$] leave will C think T C[wh$^+$]

well-formed subject wh-movement (simplified)

We can think of (28) as a record of the dominators that appear between the *wh*-mover with its wh$^-$ feature and the matching wh$^+$ feature on the C-head. As the two sentences above are well-formed, none of the intervening heads apparently affect the licensing relation between the licensee and licensor features.

Remember that the goal is for us to determine which categories can act like blockers preventing the dependency between wh$^-$ and wh$^+$. To do so, we need to contrast these well-formed *a-strings* relations against structures causing island-effects.

Now, let us put these in IBSP terms. As mentioned earlier, we need to define two edges, an environment, and some positions that specify our grammar. Since we want to establish a dependency between the wh$^-$ and wh$^+$, we can use these as our left and right edges, respectively. That means that we do not want any other wh$^+$ to appear between the two. This defines our environment. The reason for this is because if we have another wh$^+$, we are moving beyond the boundaries of a clause, and that is not what we want to evaluate. This is schematically shown below. The positions and the grammar will need to be determined for each island constraint separately. As we will see, these are going to look very similar in the end for various constraints, which is the main point of this chapter, i.e. defining a unified class for island constraints.

Figure 4.7: IBSP Template for Islands with 1 Open Slot

This is the template that I am going to use to account for various island constraints, with (re)defining the environment and the grammar. I begin with adjunct islands, move to CNPC, and other island constraints afterwards.

4.4.2 Adjunct Islands

I begin this section by looking at adjunct islands, extraction out of which seems to be banned universally (Stepanov, 2007). I then look at CNPC, which turn out to be very similar in formal terms to adjunct islands. Take the sentences in (29), and the *a-strings* relations of *which* in (29b).

(29) a. * Which topic did you leave [because Mary talked about _]? *Adjunct Island*

b. *as*(which) = which[wh^-] talk_about T because[$\text{V}^\sim$] leave T did[wh^+]

Fig. 4.8 shows the dependency tree from which the *a-strings* relations above for an AIC are computed.

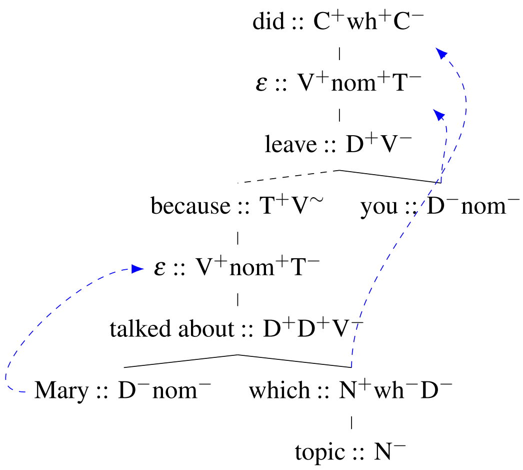


Figure 4.8: MG dependency trees of an adjunct island violation

Now, compare these *a-strings* relations with the well-formed ones in (28a) for an object *wh*-phrase, repeated below.

(30) *as*(who) = who[wh^-] leave will that[C^-] think T C[wh^+]

I mentioned earlier that the list in (28a) can be thought of as a list of the nodes that appear between the *wh*-mover with its wh^- feature the matching wh^+ feature on the C-head, and are dominated by it. In cases of well-formed sentences, the dependency between these two features goes interrupted by the nodes in between. That means that in cases of islands, there should be a node that blocks this dependency.

Looking carefully at (29) and comparing it to (28a), we can see that the blocker seems to be an adjunct having $V^{\sim}$ feature. This is in line with Sabel (2002)'s notion of "barrier", since an adjunct is a maximal projection and movement of a *wh*-phrase merged with a maximal projection is out.

Putting this in IBSP terms, using these features at the edges of the grammar, we can account for adjunct island by ruling out any containing an adjunct $X^{\sim}$ between a mover and the matching licensor feature.

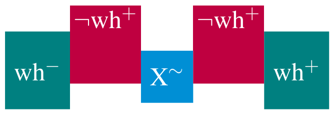


Figure 4.9: IBSP grammar for adjunct islands

Intuitively, this grammar evaluates and constraints the one open slot we have, identified with blue. However, it only evaluates this slot in the specific interval given, i.e. between a wh^- and wh^+, specified with left and right edges. Naturally, no other wh^+ should occur in between the two, otherwise, it would define a new interval on its own that needs to be evaluated. Having this interval, what the grammar forbids is an adjunct, specified with $X^{\sim}$ in the open slot. Given that the relation we have chosen is *a-strings*, what this grammar does is forbidding an adjunct that contains a wh^- and itself that is itself contained in a wh^+. In other words, it forbids *wh*-phrase movement from inside an adjunct.

There have been many accounts in theoretical syntax as to why adjuncts act as islands. We are going to see how my system can fit with some of these accounts on adjunct islands.

I already discussed Stepanov (2007)'s Late Adjunction Hypothesis, in (15), in previous sections. So, let me begin with that. This hypothesis requires adjuncts to merge postcyclically. So, for instance, in the sentence "John went to bed [after Peter fixed what]", the adjunct *after Peter fixed what* is merged after the CP *John went to bed* is built, leaving the Q feature on C unchecked and causes the derivation to crash.

One can see that my system can easily cover this proposal. The core of an IBSP grammar is defining intervals within which a grammar is evaluated. Giving the LAH, the edges cannot be established to begin with. We have a wh^- in the adjunct clause and no wh^+. Therefore, we cannot evaluate the grammar and the clause is ruled out.

I would like to evaluate my proposed grammar using yet another analysis of adjunct islands, namely Huang (1998)'s CED, in (i.), and a phase theory account of it (Chomsky, 2001).

According to the Phase Impenetrability Condition (PIC), in order to be able to extract a phrase out of a phase, its needs to be at the edge of this phase or to have moved there to be visible for the higher probe. In other words, in a configuration like [$_{ZP}$ Z ... [$_{HP}$ α [H YP]]] (Chomsky, 2001, ex. 8), only H and its edge are accessible to higher probes (4.4.2). Not every phrase constitute a phase, but CPs do (Adger, 2003), and therefore, extraction out of them can only happen on phrases on the edge of the phase.

(31) **PIC:** The domain of H is no accessible to operations at ZP; only H and its edge are accessible to such operations (Chomsky, 2001, ex.11)

We also know that successful *wh*-movements happen successive cyclically, meaning the *wh*-phrase moves to the edge of the embedded CP; it becomes visible to the higher C head that has the uninterpretable *wh*-feature that attracts it Adger (2003).

The reason that extraction out of adjuncts is ill-formed is because adjuncts are never in a position where they are θ-marked by a selecting head. By virtue of PIC, only the heads and specifiers of phrases that are θ-marked are visible to a higher phrase to trigger movement if the right feature matches via an Agree relation. Since adjuncts are never θ-marked, neither their head nor their specifier is visible to a higher phrase. As a result, movement of any *wh*-phrase out of them is illicit.

This is also easily realized using my proposed grammar. Since our grammar bans any adjunct from intervening between a *wh*-mover and a *wh*-licensor, this requirement of PIC is trivially met. I can account for this requirement in another way, which is by redefining my IBSP edges and my grammar. Instead of having wh$^+$, I can truncate my grammar to include the adjunct as its right edge, Fig. 4.10. I should then prevent any wh$^+$ to be between my left and my right edge. This grammar prevents any sentence with a *wh*-mover if it is contained in an adjunct, if no *wh*-licensor is in between. In other words, if we have a wh$^-$ inside an adjunct, it cannot move out of it.

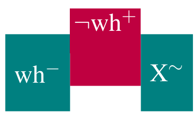


Figure 4.10: IBSP grammar for adjunct islands; Truncated Version

This procedure necessitates truncating the grammar in a way that might seem a bit ad hoc. I demonstrated it here to show the existence of this possibility. I will use this method in other island constrains to show that it is possible to make the grammar even less complex, but having matched left and right edge is more systematic and unified.

Besides having different analyses for adjunct island constraint, another point that needs some work is the fact that some adjuncts do allow for extraction (Truswell, 2007), as mentioned earlier. Two points are worth mentioning here. First, the amelioration effect has been proposed to be due to lack of tense in such constructions (Szabolcsi, 2006). Moreover, this amelioration effect is only observed in NP-extractions, and PP-extractions are practically excluded from this effect (32).

(32) a. Which topic did you leave [without talking about _]?

b. *About which topic did you leave [without talking _]? (Szabolcsi, 2006, ex.28)

Accounting for the first one, namely tenselessness, can be done using our feature system. We can specify or ban adjuncts that select for a tensed clause, such as the following, where [FIN] encodes finiteness.

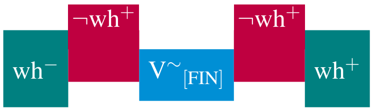


Figure 4.11: IBSP grammar for adjunct islands; encoding tensed clause

To address the second issue, namely banning PP-extraction out of infinitival clauses, our system needs to encode the fact that the wh-phrase is immediately contained inside a P, as the *a-strings* (33) for the *wh*-phrase in (32b) shows.

(33) *as*(who) = which[wh^{-}] about talking T without[V$^{\sim}$] leave T C[wh^{+}]

Taking these points into account, and keeping in mind that the feature specification of the adjunct is not going to show up in the IBSP grammar itself, we end up with the following grammar for extraction out of infinitival (Truswell) clauses.

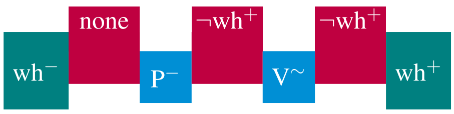


Figure 4.12: IBSP grammar for PP-extraction out of infinitival clauses

4.4.3 CNPC Effect

Using the same mechanism in the previous section, explaining CNPC is relatively easy. Complex NPs (DPs) can be divided into two groups of DPs with complement clauses (CC), repeated here in (34a), and DPs with relative clauses (RC), as in (34b).

(34) a. * What did you hear rumors [that John bought _]?

Complex NP with CC (Boeckx, 2012: 3)

b. * Which kid must you call [the teacher who punished _]?

Complex NP with RC (Szabolcsi, 2006, ex:11)

Let us first consider the CNPC with CC. The *a-string* for the *wh*-phrase in (34a) is presented in (36). The *a-string* relation for this sentence differs from the well-formed sentences in having an N that selects for a C and is contained inside a D. The corresponding dependency tree for this sentence is provided in Fig. 4.13.

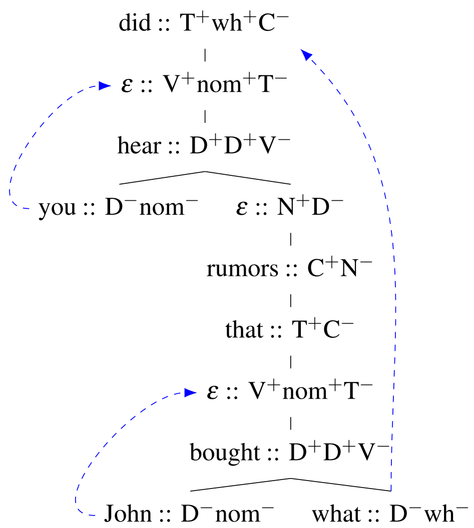


Figure 4.13: MG dependency trees of a complex NP violation

(35) $*as(\text{what}) = \text{what}[\text{wh}^-]$ bought T that rumors$[C^+]$ D you hear T did$[\text{wh}^+]$

Put in IBSP terms, this can be captured by ruling out any C containing a wh^-, that itself is immediately contained in a D (shown by the environment being set to none), between a mover and the matching licensor feature, as Fig. 4.14 shows.

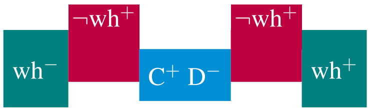


Figure 4.14: IBSP grammar for Complex NP islands

Now, let's look at the second form of CNPC with RCs. Using Kayne (1994)'s promotion analysis for RCs[4], we wind up having the *a-strings* for the *wh*-phrase as follows. The reader is encouraged to verify this result.

(36) a. * Which kid must you call [the teacher who punished _]?

Complex NP with RC

b. * *as*(which) = which[wh$^-$] punish T who ε[C$^+$ rel$^+$] the call T must[wh$^+$]

We can again see that the issue is with having a D containing a head that selects for a C and contains the wh$^-$. The fact that the same head also licenses the movement of the head of the RC does not change this observation. That being said, the same grammar proposed for CNPC with CCs holds for CNPC with RCs.

We can use the truncation strategy here as well to further reduce the complexity of the grammar. This can be done by limiting the right edge of the CNPC to D to make our IBSP be of factor 0.

Figure 4.15: IBSP grammar for Complex NP islands; Truncated

4.4.4 Sentential Subject Constraint

The sentential subject constraint is an easy illustration of another core technique for IBSP analyses. As before, we start out with a simplified *a-strings* relation for the mover. The corresponding trees can be found in Fig. 4.16.

[4]In Kayne (1994)'s analysis, the RC is selected by an external D head. The head of the RC is a nominal constituent and directly raises to the [Spec, CP] from its base position.

(37) a. * Who did [that Mary kissed _] bother you? *Sentential Subject*

b. * *as*(who) = who[wh$^-$] kiss T that[nom$^-$] bother T[nom$^+$] did[wh$^+$]

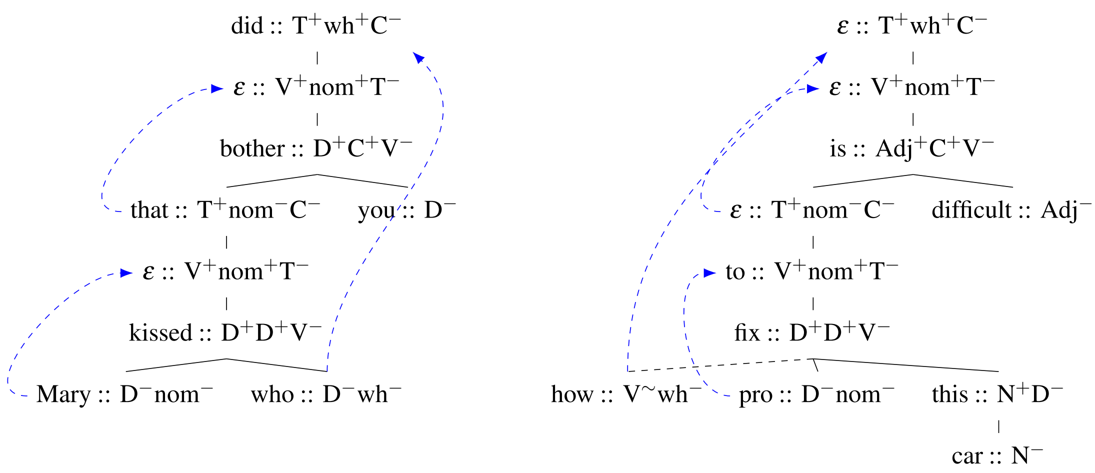


Figure 4.16: MG dependency trees with English Sentential Subject violation with object extratcion (left), and with adjunct extraction (right)

The central problem here is not so much the presence of the complementizer *that*, but rather that the whole CP containing the *wh*-mover is undergoing subject movement. Mirroring our previous approach to the adjunct and CNPC effect, we could define an interval with wh$^-$ and wh$^+$ as the edges and forbid intervening instances of that[nom$^-$].

Figure 4.17: IBSP grammar for sentential subject islands (to be modified)

I would like to work on another example of a sentential subject constraint here, in which we have a movement of an adjunct.

(38) * How is [to fix the car _] difficult? (Takahashi, 1994: 63)

The *a-strings* for the *wh*-phrase in this sentence, *how*, is obtained using the dependency tree in Fig. 4.17 (right), and is given as follows.

(39) **as*(how) = how[wh$^-$] fix T C[nom$^-$] is T C[wh$^+$]

This is exactly the same *a-string* we had for the previous example encoding object movement out of a sentential subject. Therefore, the proposed IBSP constraint holds for this example too.

Sentential subjects are not the only types of subjects that penalize extraction. Extraction out of English subjects in general causes ungrammaticality as the following examples show. (52b) shows that extraction out of subjects of passive constructions is also bad.

(40) a. * Who$_i$ does [a picture of t_i] hang on the wall? (Stepanov, 2007: 85)
b. * Who$_i$ was [a friend of t_i] arrested? (Stepanov, 2007: 80)

As one can notice, it is not the complementizer per se that movement out of which is bad. Rather, it is a nom$^-$ mover, that does not like movement of any element that it contains. Therefore, I propose the modified version of Fig. 4.17 above to account for these structures too, where X means any category.

Figure 4.18: IBSP grammar for sentential subject islands

Although this grammar works for English and might be true of other languages that have such constraint, this constraint does not hold cross-linguistically, as mentioned before. As an example, take Turkish (41a) and Russian (41b) below, languages that do not show the sentential subject constraint:

(41) a. [*Op$_i$* [*pro* [t_i anne-si]-nin herkes-le konuş-tu-ğ-u]-nu
[*Op$_i$* [*pro* [t_i mother-AGR]-GEN everyone-with talk-PST-COMP-AGR]-ACC
duy-du-ğ-um] adam.
hear-PST-COMP-AGR] man
'(lit.) The man [whose I heard [that [_ mother] talked to everyone]].'

b. S kem by ty xotel čtoby govorit' bylo by odno
with whom SUBJ you wanted that-SUBJ to-speak were SUBJ one
udovol'stvie?
pleasure
'(lit.) with whom would you want that [to speak _] were sheer pleasure?
(Stepanov, 2007: 90-91)

Stepanov (2007) provides a minimalist analysis in which the possibility of extraction out of subjects is tied to their failure of movement to [Spec, TP]. Referring to Takahashi (1994)'s analysis on Japanese, Stepanov (2007) generalizes that in such languages, the subjects remain in VP. *wh*-extraction out of such subjects targets [Spec, TP] first and is followed by another movement to [Spec, CP]. This actually falls naturally out of my proposed IBSP constraint. As Fig. 4.18 shows, the blocker in such constructions is a category with a nom$^-$ feature, which is what causes the subject movement to [Spec, TP]. If we keep the subjects in their base-positions, this nom$^-$ feature is not going to be on the subject to begin with. That means that they cannot act as blockers. Consequently, the well-formed sentences are correctly allowed. In other words, my proposed IBSP analysis blocks *wh*-movement out of a moved subject.

The IBSP grammar that I have come up is in line with the "criterial freezing" idea of Rizzi (2007) and Rizzi and Shlonsky (2007), which is a condition on movement out of moved subjects. This condition has two main components: "i) An element moved to a position dedicated to some scope-discourse interpretive property, a criterial position, is frozen in place (Criterial Freezing), and ii) Classical EPP, the requirement that clauses have subjects, can be restated as a criterial requirement, the Subject Criterion (Rizzi and Shlonsky, 2007)". This basically means that thematic subjects move to the criterial subject position and are frozen there by Criterial Freezing. My proposed IBSP grammar captures this condition perfectly. The nom$^-$ feature encodes a subject mover. And the ban itself translates into the ban against movement out of this moved subject.

So far, I have looked at three island constraints. I have discussed the exceptional cases, such as Truswell sentences as well as cross-linguistic variation in the case of sentential subject constraint. Two cases of CNPC are also covered using the same IBSP grammar. I have shown that these constraints can all be unified under the IBSP class and are at most IBSP-1. The possibility of truncating and simplifying the grammars was also explored. Table 4.1 summarizes the (original and non-truncated) IBSP constraints I have come up with up to now.

Let us move on to the next strong island constraint, namely left branch condition. Just like subject constraint, this constraint also that does not hold cross-linguistically. I will address this issue in the next section.

Constraint	IBSP Grammar	Factor
AIC	wh^- $\neg\mathrm{wh}^+$ $\mathrm{V}^{\sim}_{[\mathrm{FIN}]}$ $\neg\mathrm{wh}^+$ wh^+	1
CNPC	wh^- $\neg\mathrm{wh}^+$ $\mathrm{C}^+\ \mathrm{D}^-$ $\neg\mathrm{wh}^+$ wh^+	1
Sentential Subject	wh^- $\neg\mathrm{wh}^+$ $\mathrm{X}[\mathrm{nom}^-]$ $\neg\mathrm{wh}^+$ wh^+	1

Table 4.1: Summary of IBSP constraints for AIC, CNPC and SS

4.4.5 Left Branch Condition

The left branch condition (LBC) refers to any blocking of movement of the leftmost constituent of an NP. This blocking effect is observed in cases of extraction of determiners, possessors and adjectives out of NP (Bošković, 2005), as the examples below show.

(42) a. * Whose did you see [_ father]?

b. * Which did you buy [_ car]?

c. * That he saw [_ car].

d. * Beautiful he saw [_ houses].

e. * How much did she earn [_ money]? (Bošković, 2005, ex.1)

However, as mentioned before, this condition does not hold cross-linguistically. Latin and most Slavic languages allow for Left Branch Extraction (LBE), as (43) shows for Serbo-Croatian.

(43) Čijeg si vidio [– oca]?
whose are seen [– father]
'Whose father did you see?' (Bošković, 2005, ex.2)

I started out all the analyses by first looking at well-formed sentences comparing their *a-strings* to the ill-formed ones. The same logic holds here too. To be able to rule out ill-formed English LBCs, we first need to see how the well-formed sentences are constructed using the MG feature specifications. This is going to, as usual, be informed by theoretical findings.

Probably the most comprehensive work on LBC is Corver (1991), whose main ideas have been adapted in Bošković (2005) into the minimalist framework. The main idea is that languages which allow for LB extraction do not have DPs at all. The assumption is that movement out of DP must proceed through [Spec, DP] (Bošković, 2005). This is ruled out through the ban on adjunction to arguments, DP being one. In languages like English, with LBC, the D head acts like a barrier preventing the movement out of DP. In other words, we have the following configuration for English-type languages. Bošković (2005) proposes this for adjectival phrases, but I want to generalize it to have a PossessiveP or a DP as well. Therefore, I show this generalization using an XP as the mover.

(44) $[_{DP}$ XP_i $[_{D'}$ D $[_{NP}$ t_i $[_{DP}$ (Bošković, 2005, ex.6)

Consider the following Serbo-Croatian example with an LBE. Given that there is no DP, the adjunction requirement to [Spec, DP] is void. Moreover, NPs do not construct barriers in the minimalist program. Consequently, no violation occurs and the sentence is well-formed.

(45) Lijepe$_i$ $[_{VP}$ t_i $[_{VP}$ $[_{V'}$ gleda $[_{NP}$ t_i $[_{NP}$ kuće]]]]]
beautiful$_i$ $[_{VP}$ t_i $[_{VP}$ $[_{V'}$ watches $[_{NP}$ t_i $[_{NP}$ houses]]]]]
'Beautiful houses, he/she is watching' (Bošković, 2005, ex.19)

The dependency trees in Fig. 4.19 show the subtle contrast between Serbo-Croatian allowing LBE and English banning it. Notice the difference is in the presence (or absence) of D. The general movement f feature is used to show this type of movement as I do not want to go in the details of motivating features for such a movement.[5]

We can take the general insight here that Ds act like barriers to movement. Then, we need to ban movement of the element contained within the D. This will give us an IBSP grammar like the following.

This will correctly rule out movement of the immediately contained adjective, e.g. in the right tree in Fig. 4.19. But, for when the mover itself is a D, like *that* in the sentence

[5]There might be questions as to why DP does and VP does not act as a barrier. Bošković (2005) assumes that adjunction to XP voids the minimality barrierhood of X.

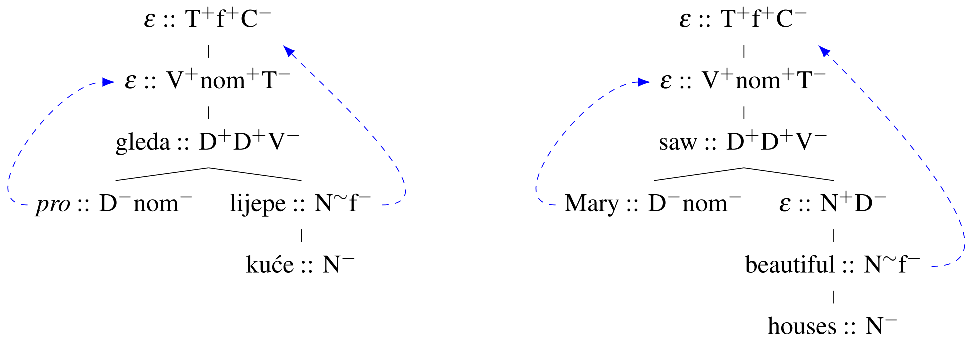


Figure 4.19: MG dependency trees with LBE for Serbo-Croatian (left), and LBC for English (right)

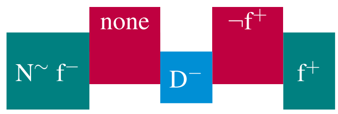


Figure 4.20: IBSP grammar for left branch condition; AP movement

(42c), see Fig. 4.21. Given that in MG, we do not show intermediate movements,[6] we lose the crucial step of adjunction to [Spec, DP] that rules out these sentences. The proposed IBSP cannot rule out the movement of D head.

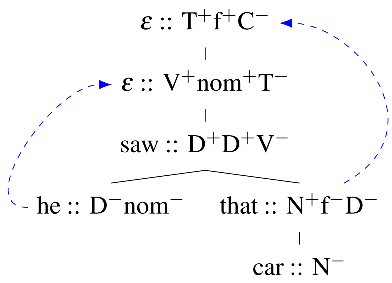


Figure 4.21: MG dependency tree for LBC for D movement in English

We need a mechanism that accounts for constituency in some way. Although not explicitly mentioned, it could have been inferred by now that in MGs any X that moves takes its projected phrase with itself. This allows us to move constituents without breaking the phrase. The issue with LBC is that part of the phrase moves (normally the head) break-

[6]This is due to the fact that each lexical item can only have a bounded number of features (Graf, 2013b: 109).

ing the constituency. To account for this illicit movement type, we need to encode head movement differently in our system.

Graf (2013b) explains that in MGs head movement, specified using a superscript $[^h]$, differs from phrasal movement in two ways. Firstly, the lexical item with a licensee feature f^- moves alone without carrying along the remainder of its projected phrase. Secondly, the landing site of this item is the head carrying the matching licensor feature f^+ rather than the specifier.

Let us see how this can help us analyze the LBC. Let's take *that*, *whose*, and *which* to be our D heads. They are specified with the $[f^{-h}]$, and $[wh^{-h}]$ features respectively, encoding that they undergo head movement. Our system needs to generally block movement of a D head with a f^{-h} feature, regardless of the distance of the movement. This can be done using the grammar in (4.22).

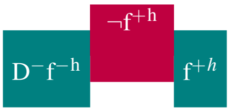


Figure 4.22: IBSP grammar for left branch condition; D movement

In general, we can ban the movement of a head that selects for another LI. This can be shown in the following IBSP diagram.

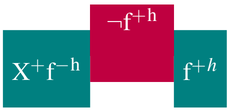


Figure 4.23: IBSP grammar for left branch condition; General

4.4.6 Section Summary

This section started out with introducing the *a-strings* representation that I adopt here to account for island constraints. This representation is basically proper dominance over MG dependency trees. What we want is to have an uninterrupted dependency relation between a *wh*-mover and its landing site (e.g. between a wh^- and a matching wh^+). However, in islands, this dependency is interrupted by a blocker. Our task is to identify these blockers and the environments they appear in.

To figure out the nature of the blockers, we need to compare the good and bad sentences. The difference between the two leads us in this direction. Using this strategy, I identified blockers for adjunct islands, complex NPs, (sentential) subjects, and finally the left branch. I showed that all these constraints can be unified under the same class of IBSP and that they are relatively simple, from being IBSP-0 in the case of left branch to IBSP-2 in the CNPC case.

The following section deals with others constraint on movement, namely the *that*-trace effect and weak islands. It is shown that these constraints, although looking similar to island constraints, are of a different nature.

4.4.7 Other Movement Constraints

This section deals with some other constraints on movement that have also received a great deal of attention in the theoretical literature. The goal is to see whether they can be grouped with the class of strong islands. I begin with *that*-trace effect and then discuss the weak islands.

4.4.7.1 That-Trace Effect

It is a well-known fact that English exhibits the so-called *that*-trace effect (Chomsky and Lasnik, 1977), which bans extraction of a subject *wh*-phrase across the complementizer *that*. The examples below compare a well-formed *wh*-movement out of an embedded clause with no overt complementizer, (46a), with an ill-formed one, (46b).

(46) a. Who do you think [_ will leave Mary]?
b. * Who do you think [that _ will leave Mary]? (Bošković, 2016: 1)

Without going into theoretical accounts of this phenomenon, I would like to determine the IBSP grammar for it first. I am going through the now familiar procedure of comparing the *a-strings* for the well-formed embedded subject *wh*-movement with the ill-formed one. These two are minimally different in the fact that in the ill-formed one the complementizer is pronounced, while the licit one has a null complementizer. The *a-strings* for the two are in (47).

(47) a. *as*(who) = who[nom$^-$...wh$^-$] leave will ε[C$^-$] think T C[wh$^+$]

b. *as(who) = who[nom$^-$...wh$^-$] leave will that[C$^-$] think T C[wh$^+$]

The simplest account of *that*-trace effect would be to rule out any overt complementizer containing the *wh* mover and the matching licensor feature, as Fig. 4.24 shows.

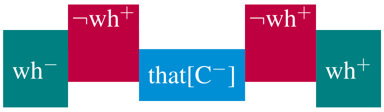


Figure 4.24: IBSP grammar for *that*-trace effect (Version 1; inadequate)

But the interval in Fig. 4.24 would also match the *a-strings* in (28) ruling out the well-formed (26a), both repeated below.

(48) a. Who do you think [that Mary will leave _]?

b. as(who) = who[wh$^-$] V T that[C$^-$] V T C[wh$^+$]

Since the *that*-trace effect does not target objects, we have to tighten the interval so that only a subject can be a valid left edge. This can be done by recourse to the MG feature calculus, which singles out subjects as those lexical items that carry nom$^-$.

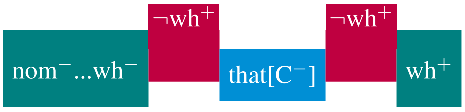


Figure 4.25: IBSP grammar for *that*-trace effect (Version 2; inadequate)

This seems to correctly rule out the ungrammatical sentence in (47b) above. However, there are cases in which the presence of the complementizer does not block movement. Such cases involve intervening adjuncts, discovered by Bresnan (1977), that ameliorate the effects of *that*-trace, as (49b) shows.

(49) a. * Who do you that [that _ will leave Mary]? that-trace effect

b. Who do you think [that under no circumstances _ will leave Mary]?

(Bošković, 2016: 1)

Adjuncts are not the only categories that cause this intervention effect. As pointed out by an examiner, even a dislocated indirect object can ameliorate the *that*-trace effect as the following example shows.

(50) John is someone who$_j$ you can be sure that on any given table$_i$ t_j will put exactly the wrong kind of silverware t_i.

We can try to accommodate this effect in our IBSP grammar, however, as the dependency tree in Fig. 4.26 shows, there is no way to include the adjunct in the *a-strings* representation of the *wh*-mover. In other words, the adjunct does not contain the wh$^-$ and therefore, cannot enter in any shape or form in our IBSP grammar. The same holds for the moved indirect object too. Given that both the *wh*-phrase and the indirect object are selected by the verb *put*, they end up being sisters to each other in the dependency tree and none contains the other. To account for this, we need to encode the c-command relations rather than dominance. But, I am actually welcoming this finding since it separates *that*-trace effect from other island constraints.

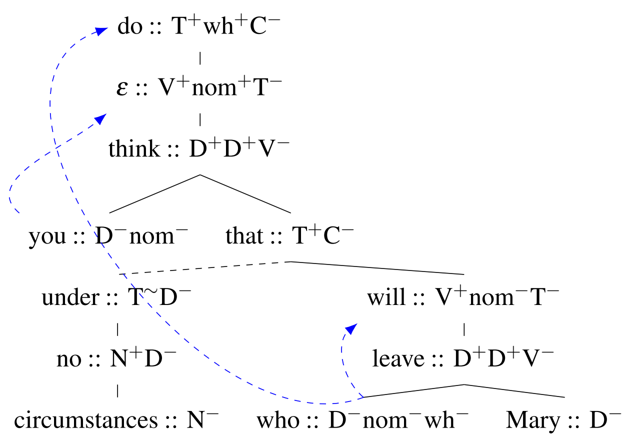


Figure 4.26: Dependency tree for amelioration effect of adjuncts in *that*-trace cases

That being said, my proposed IBSP grammar does not account for this adjunct amelioration effect. This is because the representation that I have chosen cannot incorporate this (cf. Shafiei and Graf (2020) for arguments on another type of representation). This takes the *that*-trace effect out of the realm of IBSP grammars, which could be an indicator of the fact that this effect should be kept separate from other island effects.

Before moving to the next section, I would like to discuss the cross-linguistic variation in regard with the *that*-trace effect. It is a known fact that this effect does not hold cross-linguistically. Languages such as Italian, German, Brazilian Portuguese, Greek, and Persian (to name some) do not show such an effect.

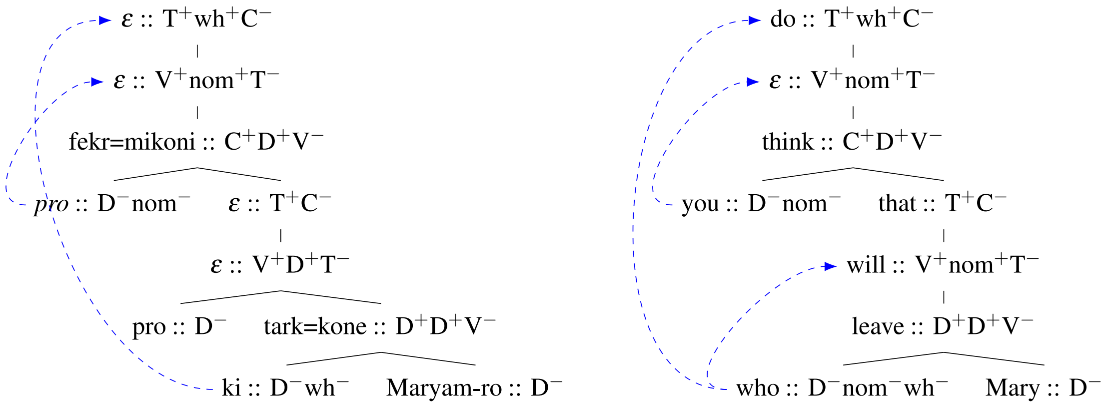


Figure 4.27: MG dependency trees with *that*-trace island violation (right), and well-fromed Persian counterpart (left)

Lack of this phenomenon has been explained by the availability of free subject inversion or the VS order, which later was shown not to hold across languages. It was then proposed by Chao (1981) that the presence of a resumptive *pro* in subject position of these structures in languages with no COMP-trace effect is what was responsible for lack of such phenomenon. Moreover, it has been proposed that in such languages, the *wh*-mover moves from a position below the [Spec, CP] unlike the English *wh*-mover.

Let us look at one of these languages in more details, namely Persian. Taking the aforementioned accounts into consideration our dependency tree for the subject *wh*-mover would look like the left tree in Fig. 4.27. To dras the tree, I have also followed Karimi (2005), who suggests that Persian subjects need not go movement to [Spec, TP]. The figure also provides the ill-formed English counterpart on the right.

Focusing on Persian-like languages with no COMP-trace effect, we can see that the *a-strings* for the mover *wh*-phrase is going to be minimally different from its English counterpart in that its features lack the nom$^-$, as (51) shows.

(51) a. Ki *pro* fekr=mikoni [*pro* ke Mary-ro _ tark=kone]? *no* COMP*-t effect*

b. *as*(who) = ki[wh$^-$] tark=kone ε ke[C$^-$] fekr=mikoni T C[wh$^+$]

The proposed IBSP grammar for this phenomenon (without taking adjunct amelioration effect into consideration), contains a nom$^-$ in its left edge on the *wh*-phrase. Firstly, this shows that this *wh*-phrase undergoes movement to [Spec, TP], which is not the case for the languages with no COMP-trace effect as their [Spec, TP] is already filled with a *pro*. Having this feature on the left edge of our IBSP grammar is sufficient to rule out English sentences without ruling out the well-formed ones in other languages[7].

Although my proposed IBSP grammar can correctly allow for movement of subject *wh*-phrase over an overt complementizer in languages with no COMP- trace effect, it can rule out the simple *that*-trace effects in English. The fact that it falls short of accounting for intervention effects might be suggestive of the idea that *that*-trace effect is a separate phenomenon and should not be classiffed together with islands. One way to think about it could be what Kandybowicz (2006) suggests, which is that the *that*-trace effect in English is a prosodic or PF-oriented phenomenon.

Kandybowicz (2006)'s prosodic analysis of the COMP-trace effect is based on evidence such as unstressed monosyllabic complementizers and the contraction of a C head with the auxiliary verb across a trace. The following examples illustrate these points.

(52) **Unstressed Monosyllabic C**

a. Who do you think *that/?th't _ wrote Barriers?

b. Who do you hope *for/?fer _ to win? (Kandybowicz, 2006, ex.5)

(53) **Contraction Across a Subject Trace**

a. ?Who do you suppose that'll leave early?

b. ?The author that the editor predicts that'll be adored.

c. ?It was John that the author told us that'd plagiarize her book.

(Kandybowicz, 2006, ex.10)

He proposes that this phenomenon is basically due to a restriction of the <C, trace> sequence in the phonological domain. For him, this sequence is illicit if it is within the same prosodic boundary. In other words, if we shift the nuclear pitch accent to the verb, for

[7]One point that needs clarification here is the feature motivating the overt movement of the *wh*-phrase in wh-in-situ languages discussed in this paper, Persian being one. Given that the overt movement is optional in such languages, one might wonder the feature system for such movement should be different from English *wh*-movement, which is indeed the case. However, unlike theoretical syntax which posits a completely new feature for such movement, such as Σ (Kawamura, 2004), we can use the Movement-Generalized MGs (Graf, 2012b), which allows us to encode *overt* versus *covert* movement types in our system. Not going into formal definitions, the intuition behind MGMGs is that every lexical item consists of features including the feature OVERT whose values is either *o* for *overt* or *c* for *covert*. For the purposes of space, I do not show all the features in my trees, however, for wh-in-situ languages with covert *wh*, this *covert* movement feature is assumed.

instance, the grammaticality is going to increase to a great extent, as the contrast between the sentences in (54) shows:

(54) **Contraction Across a Subject Trace**

a. *[Who did you say [$_{intP}$ that _ wrote Barriers]]?

b. ?[Who did you say that] __ [$_{intP}$ that _ WROTE Barriers]?

(Kandybowicz, 2006, ex.14)

The inability of my proposed IBSP grammar to account for *that*-trace effect in general is in line with the PF accounts of this phenomenon (please see the references within Kandybowicz (2006) for such accounts). There seems to be determinants beyond syntactic factors that come into play and give rise to this effect. Another phenomenon that does not quite fit the IBSP class is the notion of weak islands which is discussed in the next section.

4.4.7.2 Weak Islands

Weak islands compose of a large class of constituents, extraction out of which depends on the nature of the mover (Huang, 1998; Rizzi, 1990; Szabolcsi, 2006; Boeckx, 2012). Examples are given for *wh*-islands and *negative*-islands.

(55) a. ??Who/Which girl do you wonder [whether Bill kissed _]?

b. *How/In what way do you wonder [whether Bill kissed Sue _]?

c. Who don't you think that John talked to _? (Miyagawa, 2004, ex.4a)

d. *Why don't you think that John talked to Mary _? (Miyagawa, 2004, ex.4b)

Perhaps the most intriguing theory of weak islands is Rizzi (1990)'s (and subsequent works such as Rizzi (2001, 2004)) Relativized Minimality (RM), which aims to provide a unified account of weak islands under an intervention effect. The core idea in Rizzi (1990) is that wo syntactically related elements can tolerate no intervening element of the same kind.

Based on this account, a local relation between an operator and its variable is blocked by a third intervener, as far as it is sufficiently similar to the operator. According to RM, in the configuration "...X...Z...Y...", Y cannot be related to X if Z is an element of the same kind. More specifically, X cannot govern Y if Z is a typical potential governor of Y, where government can either be head-government or antecedent-government. Antecedent government comes in three flavors, whereby in A–chains the typical potential governor is an A-specifier c-commanding Y; in A′-chains it is an A′-specifier c-commanding Y and in X^0 chains it is a head c-commanding Y.

A specific application of the more articulated theory of Relativized Minimality proposed in Rizzi (1990) is what is called "featural Relativized Minimality" (fRM). The fRM makes use of more fine-grained feature classes classes of the elements involved in the movement and the potential blockers. The more feature match between the mover and the intervener, the more severe the ungrammaticality of the move. This way, one can explain the differences in the degree of acceptability in the case of, for instance, *wh*-islands. Consider the following contrasting examples from Italian, which suggest that the presence of the lexical restriction in the moved *wh*-phrase is critical.

(56) a. Quanti problemi non sai come risolvere?
'How many problems don't you know how to solve?

b. *Quanti non sai come risolverne?
'How many don't you know how to solve of-them?

(Friedmann et al., 2009, ex.25)

Friedmann et al. (2009) assume that there is a difference between lexically restricted *wh*-phrases and bare *wh*-phrases. While the former are attracted by a complex attractor [+Q, +NP], with "NP" designating the lexical restriction, bare *wh*-phrases are attracted by a pure [+Q] attractor. Applying this to the examples above, we can see that the *wh*-mover in (56a) contains both +Q and +NP features while the intervening *wh*-phrase only has the +Q feature, and so the blocking effect is mild (or none). The case of (56b) is worse since both *wh*-phrases have the +Q feature and so, the feature overlap is absolute and hence, the severity. We can show this in the configuration below.

(57) a. **Quanti problemi [+Q +NP]** non sai **come [+Q]** risolvere?

b. ***Quanti [+Q]** non sai **come [+Q]** risolverne?

The basic idea is that we need an account of the gradability of ungrammaticality of movement out of various constituents and fRM seems to do that. However, I cannot say the same for my proposed IBSP grammar. The IBSP grammar is set in a way to either completely rule in or to rule out configurations. What it means is that, while we can probably engineer it in a way to rule out, let's say, extraction of adjuncts out of *wh*-islands, we will not be able to account for the more acceptable object extraction out of the same constituents. We can play with the features to account for other types of weak islands, but yet again, we will not be able to account for the marginal cases.

There has been suggestions in the literature that weak islands should be explained in semantics and/or pragmatic terms (as suggested in Szabolcsi (2006) and reinforced in Boeckx (2012)). This explains the lack of uniformity among the class of blockers as well as the failure of the IBSP class to account for gradability in this group of constraints. Put it another way, the fact that the IBSP class is absolute suggests that what is called a weak island is not a purely syntactic phenomenon.

To capture *gradience* in phonology, Mayer (2021) defines probablistic TSL (pTSL) grammars as an extension of the TSL class. He shows that the TSL class is a proper subset of the pTSL class, meaning that while we can define a pTSL grammar for a given TSL language, the opposite does not necessarily hold. Using such grammars, he can account for the distance-based decay in phonological dependencies, where there is a negative correlation between the the amount of irrelevant material between relevant segments and how strongly the dependency holds. Some dependencies he looks at are Hungarian and Uyghur vowel harmony. Mayer (2021)'s pTSL grammars, although not free from deviation, greatly mirror the results of online rating tests. He argues that the pTSL grammar can be extended to handle contextual projections in other TSL extensions such as IO-TSL. In fact, he argues that some of the deviations observed in his pTSL grammars can be avoided using contextual projections. What is worth noting is that the probablity is defined as the likelihood of each segment being projected to the tier, whereas the defined grammar is inviolable. Whether or how this pTSL class can be used to account for gradience in the case of *weak islans* is beyond the scope of the current work, but something worth pursuing in the future.

4.5 Formal Discussion and Linguistic Implications

I was able to show that islands can be expressed as IBSP constraints on a specific kind of path language for MG derivation trees. Each path indicates a node's *a-strings*, which in turn gives us the list of nodes containing or dominating that node. Once that is established, an IBSP analysis is fairly straight-forward. It encodes the nature of the blockers preventing movement from among the items they contain.

It is striking that all island constraints are variations of a simple pattern. The left edge picks out a mover, and the right edge identifies a licensor. Keeping the right edge to identify the licensor, the most complex pattern we have is IBSP-2, for CNPC effect, which suggests a surprising degree of computational parallelism between phonological dependencies and syntactic dependencies.

While my findings still need to be vetted by a more detailed and typological analysis of a much wider range of constraints and across many languages, it is encouraging that they closely mirror those in phonology and yield rigorous claims about what is a possible island.

That said, I am fully aware that the IBSP perspective does not tell the full story. Some aspects of islands can be accounted for, but not in an insightful manner. For instance, in weak islands, the acceptability of extraction can be affected by whether the mover is an argument or an adjunct. While one could address this by introducing a weighted version of IBSP that punishes adjunct movement more than argument movement, it would only restate the facts. One could just as well set up the system the other way round so that adjunct movement is more acceptable than argument movement, yet no known language seems to behave this way. Hence IBSP can tell us something about the overall arrangement of the elements that make up an island interval, but it is agnostic about the syntactic substance of these elements.

For the same reason, IBSP is incapable of generalizing across different movement types. An adjunct is an island for both *wh*-movement and topicalization, but these require distinct number of open slots; one with wh^- as the left edge and no wh^+ among the fillers, and one with top^- as the left edge and no top^+ among the fillers. Strictly speaking, then, islands constraints require multiple IBSP-grammars, one for each movement type. Why some movement types cluster together, although interesting, is orthogonal to the issues that can be explored with IBSP.

What is more is that IBSP is limited and can apply to only one clause. That being said, it falls short of evaluating the phenomena that require Across The Board (ATB) movement, including Parasitic Gaps (PGs) and the Coordinate Structure Constraint (CSC). The CSC forbids extraction from a conjunct.

(58) * Which wine did [Ed brew beer and Greg drink _]?

While it is simple enough to block extraction across the head of a coordination phrase, the problem is that the constraint is suspended if every conjunct contains a gap.

(59) Which wine did [Ed brew _ and Greg drink _]?

Since there is no dominance relation between these gaps, neither one appears in the other's *a-strings* list. The *a-strings* for the objects of *drink* do not differ at all between the two sentences. Therefore, IBSP does not provide all the answers, and alternative perspectives such as the algebraic approach of Graf (2013b) and *subregular* syntax of Graf (2018b) are still needed. But this does not diminish the value of the linguistic insights provided by IBSP.

Another exception that IBSP falls short of explaining is the English cases for which the sentential subject condition does not hold as (60) shows. The IBSP grammar I have proposed for sentential subject includes a complementizer as blocker, however, there is no such blocker in the dominance relation of the *wh*-phrase in (60).

(60) Who is there [a picture of _] on the wall? (Stepanov, 2007, ex.31)

Last but not least, the details of how IBSP can account for other blocker types, such as topics, negation markers, certain quantificational elements, examples of which are below (Rizzi, 1990), are yet to be carved out.

(61) a. * What did John say [that to Bill Sue gave _]?
b. * I asked [how John did not behave].
c. * How did only John think [that you behaved _]?
d. * How did you deny [that you behaved _]?

That being said, the IBSP grammar can generalize over the class of possible phenomena, and also explain the absence of several conceivable island constraints.

To begin with, the IBSP grammars each apply individually. Once a domain is instantiated, the sentence is ruled out. This would ensure that the distance between the mover and the blocker does not really matter. This is observed in sentences (62a) and (62b), which are correctly ruled out by the IBSPs for AIC and CNPC, irrespective of the intervening infinitival phrases (remember infinitival phrases alleviate island effects). This is because the IBSP grammars each apply individually. Once a domain is instantiated, the sentence is ruled out. In (62a), the domain for AIC is instantiated once we hit the adjunct *because*. For the same reason, the sentence in (62b) is also ruled out when the domain for CNP is instantiated. These are irrespective of the fact that the *wh*-mover is contained within an infinitival clause. For this reason, (62c) is correctly ruled in because none of the IBSP domains are instantiated.

(62) a. * Which topic did you leave [because you discussed [talking about _]]?

b. * Which topic did you leave [without [talking about the claim that John liked _]]?

c. Which topic did you leave [without [taking about discussing _]]?

In other words, the IBSP constraints are not monoclausal. What this means is that they are not limited to the clause containing the blocker and apply across clauses (as long as the edge relation wh^{-} and wh^{+} holds). Take (63) for example with two clauses intervening between the adjunct blocker and the *wh*-gap[8].

(63) * What did you like [because John said [that Mary had told the professor [to discuss _]]]?

Building on this discussion, we can also preclude patterns such as a variant where extraction is possible only if the number of nodes dominating *that*-complementizers is odd. Even without the additional restrictions on intervals, this island constraint would be inexpressible because IBSP is even weaker than first-order logic, which cannot do *modulo* counting.

Not only can IBSP generalize over some attested island constraints, it can overgeneralize as well. Imagine having an IBSP-0, similar to the grammar proposed for LBC. Such a grammar explains some natural language phenomena, such as anti-locality (64), which has been proposed by Bošković (2016) as an account for some island constraints such as CNPC and LBC as well as the *that*-trace effect.

(64) Antilocality: Movement of A targeting B must cross a projection distinct from B (where unlabeled projections are not distinct from labeled projections).

(Bošković, 2016, ex.30-31)

Take his analysis for *that*-trace effect for example, which can be accounted for using a grammar such as Fig. 4.28.

(65) a. * Who$_i$ do you think that t_i left Mary?

b. Who$_i$ do you think [? t_i [$_{\mathrm{CP}}$ that [? t_i [$_{\mathrm{IP}}$ left Mary]]]]?

(Bošković, 2016, ex.30-31)

Figure 4.28: IBSP grammar for antilocality

This grammar blocks the movement of a mover with wh^- if there is nothing (no phase in the sense of Bošković (2016)) between it and its licensor, wh^+. Although this grammar can correctly account for antilocality, it can also result in patterns that are not naturally possible. Take a look at a minimally different grammar provided below:

Figure 4.29: IBSP grammar for movement across a special node

This grammar requires a mover to move across a special node, here C. While this is formally possible, and there might be some cases in natural languages, e.g. movement of a subject to [Spec, TP] should cross a *v*P, if I complicate things a bit, it loses its naturalness. Take the following as a possible formal grammar:

Figure 4.30: IBSP grammar for movement across special nodes

This grammar can enforce movement across one of these but not two. Let's say we move f^- across a V, then you cannot have a C on the way of the mover and vice versa. In general, if we have IBSP-n, we can enforce movement across n+1 nodes. This is something that is not attested in any natural language. In sum, as I showed IBSP can capture different island constraints together with some cross-linguistic variation. But in some cases it falls short of explaining attested patterns, while in some other cases it overgeneralizes and is too powerful.

[8]It has been noted in the literature that as the length of the dependency increases, the acceptability rate decreases (Sprouse et al., 2012).

4.6 Conclusion

I have shown that island constraints can be expressed as, at most, IBSP-2 constraints on a particular kind of path language that encodes dominance or ancestor relations on a dependency tree. The limitation to IBSP-2 tightens the linguistic typology by excluding logically conceivable yet unattested island constraints.

This is yet another step towards reducing the complexity of syntactic dependencies in an effort to put them in the realm of phonology, which has been shown to be fairly *subregular*. Having reduced the island complexity to IBSP-2 is in a way in line with McMullin (2016), which shows phonological dependencies are at most TSL-2.

While many island constraints can also be expressed in the *subregular* tree-tier system of Graf (2018b), the two diverge with respect to the patterns they rule out. I hope to unify those two approaches in the future in order to give an even tighter and linguistically insightful characterization of island constraints. Moreover, using Graf and Shafiei (2019)'s *c-strings* would also be able to account for these effects in the sense of IBSP grammar, but using that system would increase the complexity and we would end up having IBSP-3 to address these constraints. Using *a-strings* instead is also a way to differentiate island constraints differ from other types of syntactic constraints discussed in (Graf and Shafiei, 2019).

We saw in Chapter 3 that some syntactic dependencies, such as reflexive binding, fall under a particular subclass of the TSL languages, namely the IO-TSL languages. The findings of this chapter act as a reassurance and support for the findings in the previous chapter.

Chapter 5

Conclusion

This thesis investigates two syntactic dependencies from a computational and formal perspective. The objective of this investigation is threefold. Firstly, to show that using a formal viewpoint, different dependencies form a natural class in that they fit into one formal class. Secondly, to illustrate that given the right representation, syntactic non-local dependencies are *subregular*. Lastly, to show that the computational complexity of non-local dependencies is the same across domains and that syntactic operations are in line with phonological and morphological operations. A nice side finding of the current work is that relativized locality plays a major role in defining the complexity of different dependencies.

To be more specific, I define a specific path language over MG dependency trees to be used as in intermediate representation in lieu of tree representation. I call this path language *c-strings* to represent c-command relations, and *a-strings* to represent dominance. Using *c-strings*, I show binding conditions fit the IO-TSL extension of the TSL class. Furthermore, I show that using *a-strings* allows us to classify island constrains under the IBSP extension of the SP class. Both these classes are within the *subregular* class of languages.

5.1 Unified Classes

Theoretical linguistics has made great contributions to our understanding of syntactic long distance dependencies. The extensive and diverse work of scholars on the areas of binding and island constraints has not only given birth to new and innovative theories, but also has enriched the dimensions of the empirical datapoints across a variety of languages and

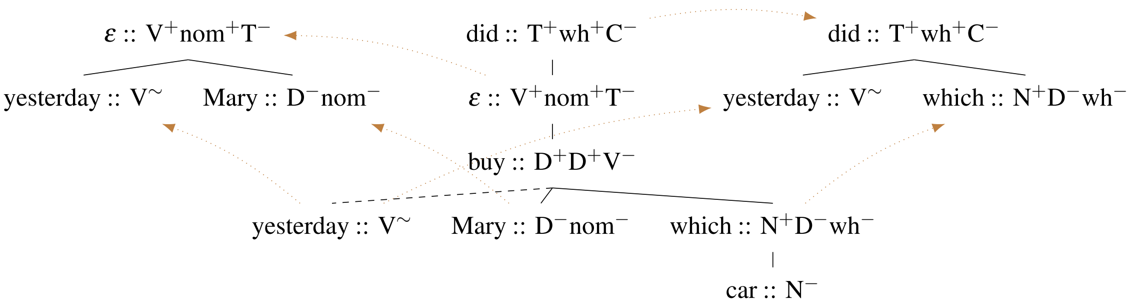


Figure 5.1: Well-formed tier projections for a *nom*-movement and *wh*-movement.

language families for such operations. Nevertheless, the need and the desire to consolidate the data and theories into one elegent analysis is still lacking in the theoretical world. This thesis is the first step in addressing this issue in an attempt to provide a unified class which these dependencies belong to. I show that to achieve this, we need to change our perspective, and move beyond the purely theoretical view. Moreover, I show that using FLT as our new lens, bulit upon empirical and theoretical findings, this unification is possible.

As outlined in §2.2 for *c-strings* and §4.4.1 for *a-strings*, I use intermediate path languages as my representations to model these dependencies. Using string representations, I show that binding conditions fit into the IO-TSL class, while island constraints can be explained in the IBSP class. That being said, while I used different *subregular* classes to account for these two phenomena, one can argue that they might fit into other language classes as well. For instance, one can show that island constraints are TSL over trees. Remember from Chapter 2 that *Move* is TSL over MG derivation trees (Graf, 2018b). The same holds for MG dependency trees as Fig. 5.1 shows.

However, the tier projection for *Move* is ill-formed for, let's say, an Adjunct Island Constraint (AIC). This is because the two conditions for *Move* tiers are not met here. Those conditions require that every *Move* node have exactly one LI among its daughters, and very LI have a *Move* node as its mother. However, as Fig. 5.2 shows this mother-daughter relation is interrupted by another LI in between.

The same generalization holds for all other island constraints. The idea is that islands are a natural corollary of having the basic syntactic operation of *Move* under the class of TSL languages (Graf, 2018b). This is due to the fact that the result of the *Move* tier projection for islands is illicit. While for the well-formed sentences, we would want the tier to only consist of the mover and its landing site. The *Move* tiers for islands will have

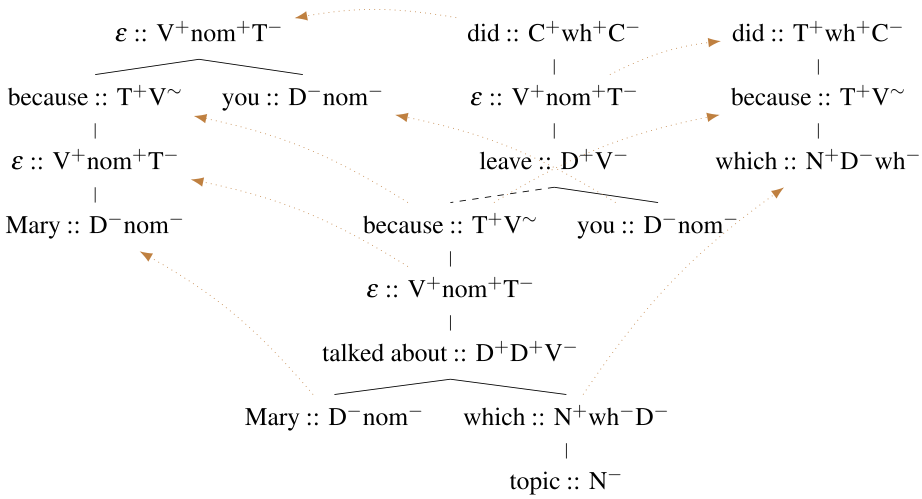


Figure 5.2: *Move* tier projections for AIC

intervening nodes in between. From this perspective, islands form a natural class, not conceivable otherwise. One can conclude that the cognitive ability for movement entails the cognitive ability for islands.

I used this example to show that it is not the case that “one size fits all”, and the ability to explain a phenomenon in one class does not deny its capability to fit another class.

5.2 Cognitive Parallelism

Earlier works in phonology and morphology show that such operations are *regular* string languages (Johnson, 1972; Kaplan and Kay, 1994; Karttunen et al., 1992). Syntax seems to have been the odd one out as it can be shown to be *mildly context-sensitive* over strings or yield of trees (Shieber, 1985). In other words, while phonology and morphology were proposed to fall on the far left of the complexity spectrum, syntax occupied a higher up place in the hierarchy. However, this idea or claim that syntax is radically different from phonology and morphology does not hold for at least some aspects of syntax investigated in this thesis. I showed that some operations or phenomena are regular once we derive the right intermediate representations from trees. This puts syntax in parallel with phonology and morphology, which by themselves are proven to be even less complex than the *regular* language class.

Recent works in the fields of phonology and morphology show that these dependencies can be modeled in a less complex fashion, and that in fact they occupy a space below the class of *regular* classes, hence, the name *subregular*. Numerous papers, such as Heinz (2010); Heinz et al. (2011); Jardine and Heinz (2016); Heinz (2018), to name some, show that phonotactics and phonological dependencies are *subregular*. Moreover, they show that there is an upper-bound for their complexity that can be captured by a specific *subregular* class of Tier-based Strictly Local (TSL), and its extensions (Graf and Mayer, 2018). Works on morphology, such as Chandlee (2014); Aksënova et al. (2016), to name some, also reach similar conclusions, showing that morphological dependencies are also *subregular* over strings.

Syntax is known to be *mildly context-sensitive* over strings. However, it seems to be at most *regular* over trees. The research on *subregular syntax* has received some attention in the past years, through the works of Graf (2012a, 2018b); Graf and Heinz (2015), among others. These works focus on the fundamental operations of *Merge* and *Move* in the framework of Minimalist Grammars (Stabler, 1997, 2011). They show that these operations are either SL or TSL. More specifically, they show that *Merge* without adjunction is SL, while *Merge* with adjunction and *Move* are both TSL. These analyses use MGs derivation tree representations as their fundamental relation, and the findings are based on the feature properties of the lexical items.

Among other work that use trees as their representations, one can name those of Vu (2018, 2020). These works inspect an empirical phenomenon, namely NPI licensing, and show that this dependency can be described with Input-local Tier-based Strictly Local (I-TSL) or Multiple Input-local Tier-based Strictly Local (MITSL) restrictions, and basically, are *subregular* over trees. Vu et al. (2019), on the other hand, work on another linguistic dependency, namely the Dependent Case Theory, and model it using TSL class.

What they all have in common is the use of trees for their representation. But, trees themselves add another level of abstraction to the equation. The more established works on phonology and morphology use string representations as their base format. The use of strings is well-established and the findings for strings can easily be lifted to other representation forms (Heinz, 2018). Therefore, I use a special form of a path language in this thesis to encode various syntactic relations and dependencies.

5.3 Limitations

While current work establishes a solid enough foundation for studying further syntactic dependencies, its scope is limited. For instance, the binding relation that I looked at in Chapter 3 focuses on reflexives and their licensing requirements. However, we know that there are other binding conditions, that regulate the behavior of pronouns and R-expressions, i.e. Princinple B and Princinple C, respectively.

In the preliminary sense, the behavior of pronouns is complementary to the behavior of reflexvies. Principle B requires that non-reflexive pronouns not be bound within their local domain. This is the mirror image of Principle A. The following examples show this complementary distribution between reflexive and non-reflexive pronouns.

(1) a. Peter$_i$ watches himself$_i$/*him$_i$ in the mirror.

b. Carlotta$_i$'s dog accompanies her$_i$/*herself$_i$ to kindergarten. (Büring, 2005: 6)

Given the boundedness of the domain that the pronoun needs to be free in, a similar mechanism to reflexive binding can be utilized here as well. The *c-srtings* of the pronoun is well-formed if a potential binder is beyond the limits of the its binding domain, and is ill-formed otherwise. Using this logic, one can say that in its preliminary form, Principle B can be captured by the IO-TSL grammar.

However, it is known that Principles A and B are not always in complementary distribution. See the following example for picture-phrases. In these examples, for Principle A, the local domain of picture-phrases includes the c-commanding arguments of the verb and the posessor whereas whereas for Principle B, this local domain only includes the possessor:

(2) a. John$_i$ finally saw Mary's picture of himself$_i$/him$_i$.

b. Mary finally saw John$_i$'s picture of himself$_i$/*him$_i$.

(Champollion, 2008, ex. 4-5)

When there is no posessor in the picture-phrase, we also see a similar distinction between the two conditions, where the local domain of the picture-phrase complement includes the c-commanding arguments of the verb for the purpose of condition A, but not condition B.

(3) John$_i$ found a picture of him$_i$/himself$_i$ (Champollion, 2008, ex. 6)

These differences in behavior between Principle A and Principle B might complicate how Principle B is modeled in my framework. However, this might be an indication in the right direction that the behavior of pronouns (at least in English) is dictated by semantics rather than syntax, and example of which is Reinhart and Reuland (1993), where principle A is defined on syntactic predicates and principle B on semantic predicates (see more references within Kiparsky (2002)).

Principle C is a bit different, on the other hand. There is no boundedness to the domain in which it applies to. This principle requires R-expressions to be free everywhere.

Kiparsky (2002) has carved out a nice typhology of pronouns, as the following figure illustrates. He proposes four binary divisions for this hierarchy. He argues that referentially independent pronouns are necessarily non-reflexive, whereas referentially dependent pronouns may be *reflexvie* or *non-reflexive*. More fine-grained difference between these two groups of pronouns is that while reflexive pronouns need a syntactic antecedent, non-reflexive pronouns can (but need not) get their reference from context/discourse. More importantly, reflexive pronouns may either be *finite-bound* or *non-finite-bound*, whereas bon-reflexive pronouns are necessarily non-bound. Finite-bound pronouns require an antecedent within the same finite clause, and may be subject to the requirement that they be *locally bound*. Such pronouns require an antecedent in the first accessible subject domain, whereas non-locally bound pronouns can (but need not) have a "long-distance" antecedent.

referentially independent | referentially dependent | reflexive | finite-bound | locally bound

Figure 5.3: Kiparsky's typology of pronouns (Kiparsky, 2002, ex.59)

Given this hierarchy and the binary division between different pronoun types, this thesis provides a substansive computational analysis of the first three classes in the hierarchy with some suggestions to expand the analysis to encompass the larger class of "referentially dependent" class. However, there is still much work to be done to understand the "non-reflexive" pronouns within the same class.

Apart from the coverage or lack-thereof for all pronoun types, what is more is that adding movement tends to complicate the analysis. While the obtained *c-strings* can be modeled in the same language class of IO-TSL, it is not clear how complex the introduction of copies to the *c-strings* is. The arbitratration strategies between different *c-string* representations for the same reflexive that is bound via movement, be it its own movement or its licensor's movement, remain to be addressed.

In regards to the island constraints, the current work provides a unified account of strong islands. However, weak islands and other movement constraints such as the *that*-trace effect do not rightfully fit the class of IBSP. I proposed in Chapter 4 that given that such constraints act differently, one should not group them together under the name of island constraints. I proposed that they differ in nature from the island constraints. More research might reveal otherwise.

It also remains a mystery why some constructions, such as parasitic gaps or Across The Board movement do not seem to fit to any of these language classes (Laszakovits and Graf, 2020). Dependencies that go beyond c-command cannot be handled with our approach. But, why that is the case requires more work. It remains to be seen whether they can be accommodated with tree tiers as proposed in Graf (2018b), or whether a completely new perspective is needed for these phenomena. A more careful examination and/or a different theoretical account of such operations might reveal that in fact they can be accounted for using path constraints, similar to what has been used in this thesis.

5.4 Future Work

I would like to start this section by saying that the future looks bright. Although the field is relatively new and there is still a long road ahead, the early works, current thesis included, are very promising. The limitations mostly come from the lack of empirical coverage, which is understandable given the recency of the research area, and it is something that can easily be addressed and added to the literature.

This thesis marks but a first step towards a *subregular* theory of syntactic dependencies, and a lot remains to be done. It does not stop there though. It is introducing a new (intermediate) string representation for syntactic dependencies. Although I mentioned my aim is not to replace tree representation with string-based syntax, the question that remains is what happens if we do? My work in "stringifying" syntax is not new. Perhaps, one of the early attempts in representing syntax with strings dates back to Kornai (1985). In his work, Kornai (1985) argues that the claim that syntax is *mildly context-sensitive* is due to the assumption that languages allow for arbitrarily long self-embeddings. What this does not take into account is that human physiology and brain make-up can only comprehend a limited number of self-embeddings. He argues that syntax is counter-free, and therefore, *regular* if we ignore the distinction between competence and performance.

The interesting connection that this early work has with the current work is that the idea of using string representation for syntax has its appeals. However, there is a major difference between his approach and my approach here, which is that he is looking at the yields of strings while I look at the string representation of specific dependencies. While for him, in principle, we can have an infinite number of embedded clauses that push the complexity beyond the *subregular* realm, we do not face that issue here. It is because the relation between the two elements can be encoded by finite means. In other words, there is no need to appeal to the performance-competence distinction, or lack thereof, as the size of the dependency is already limited in my system. Given that, I believe such a mechanism can prove fruitful if applied to other syntactic operations. If we can reduce the size of dependency and localize it, we can re-define the complexities for such dependencis.

Once the dependencies are defined in a simpler way, defining new learning algorithms would be easier. Similar algorithms have been proposed for various classes for phonological dependencies (see García and Ruiz (2004); Heinz and Rogers (2013); Jardine and Heinz (2016); a.o.). These learning algorithms can be used to inform machine learning and/or natural language processing techniques. Aksënova (2020), for instance, uses such algorithms and successfully runs learning experiments for various *subregular* classes. She shows that these languages are learned in the limit of positive data, in the sense of Gold (1967).

Lastly, the current approach only measures the complexity of a syntactic dependency with respect to a specific node. To check the whole dependency tree, one has to evaluate the *c-string* of each node. I do not know whether the TSL-approach of Graf (2018b) provides a method for doing so. As long as all dependencies are IO-TSL, though, the well-formedness of the whole dependency tree can be verified by a deterministic top-down tree automaton with a look-ahead of 1. This implies subregular complexity and may even allow for highly efficient parsing algorithms.

Appendix A

Appendix: Formal Definitions

This appendix provides formal definitions fot the relevant *subregular* classes for this thesis.

A.1 Strictly Local (SL)

The class *SL*, formally defined in Definition 3, captures local dependencies, by looking at adjacent segments. In the definitions, $\rtimes$ and $\ltimes$ mark the left and right edge of a string, respectively.

Definition 3. A *language L* is is Strictly k-local (SL-k) *iff* there exists a finite set $S \subseteq F_k(\rtimes\Sigma^*\ltimes)$ such that $L = \{w \in \Sigma^* : F_k(\rtimes w \ltimes) \subseteq S\}$, where the symbols $\rtimes$ and $\ltimes$ denote left and right word boundaries. (Heinz et al., 2011: 59)

In this definition, F_k is the set of *k-factors* of w, which is a substring of w of the length k. In other words, a string u is a *factor* of w if there is an x and y in the alphabet such that $w = xuy$. The set of *k-factors* for a given word w is defined below:

Definition 4. $F_k(w)$ = {u: u is a k-factor of w}. (Heinz et al., 2011: 59)

As Heinz et al. (2011) mention, a language is said to be Strictly Local (SL) *iff* it is SL_k for some k. For example, if our alphabet consists of *a, b, c* symbols, and our language is defined as $L = aa^*(b+c)$, we can show that L is SL_2. This is because we can define $S = \{\rtimes a, ab, ac, aa, b\ltimes, c\ltimes\}$, and for every $w \subseteq L$, every 2-factor of $\rtimes w \ltimes$ belongs to S.

We can define SL grammars in two ways, either using a positive set of factors, *permissible* factors, or a set of negative or *forbidden* factors. The grammar S in the previous example denotes a set of *permissible* factors. For instance, in the example above, $\rtimes b$ and *bb* would be in the set of forbidden factors for our given language. It is worth noting that the set of *forbidden* factors is finite.

A.2 Strictly Piecewise (SP)

The class SP, formally defined in Definition 7, captures non-local dependencies without blocking effect, by looking at subsequent segments. If we define *subsequence* relation below in Definition 5, and use the $\sqsubseteq$ symbol for it, then P_k denotes the set of such subsequences for a given word w, as defined in 6.

Definition 5. Subsequent Relation:
$w \sqsubseteq v \Leftrightarrow w = \varepsilon$ or $w = \sigma_1 \ldots \sigma_n$ and $(\exists w_0, \ldots, w_n \in \Sigma^*)$ $[v = w_0\sigma_1 w_1..\sigma_n w_n]$.

Definition 6. $P_k(w) = \{v \in \Sigma^k \mid v \sqsubseteq w\}$ and $P_{\leq k}(w) = \{v \in \Sigma^{\leq k}(w) \mid v \sqsubseteq w\}$

Definition 7. A SP_k grammar is a pair G= $\langle \Sigma, T\rangle$ where $T \subseteq \Sigma^k$. The language licensed by a SP_k grammar is L(G) $\stackrel{\text{def}}{=} \{w \in \Sigma^* \mid P_{\leq k}(w) \subseteq P_{\leq k}(T)\}$.
A language is SP_k iff it is L(G) for some SP_k grammar G. It is SP iff it is SP_k for some k.
(Rogers et al., 2009)

In other words, "A Strictly k-Piecewise (SP_k) stringset is one which can be defined as the conjunction of negative literals, where the literals are interpreted as subsequences, and whose longest forbidden literal (subsequence) is of length k. The Strictly Piecewise stringsets are those that are SP_k for some k (Heinz, 2018: 20)".

A.3 Tier-based Strictly Local (TSL)

The class of TSL languages, originally defined in Heinz et al. (2011), as in Definition 8, is a means of making non-local dependencies local by putting the relevant elements on a tier. This mechanism is in particular useful, and distinct from a similar mechanism utilized by the SP class, in that it can be used to model non-local dependencies with blocking effects. With F_k defined in 4 above, a TSL language L can be defined below.

Definition 8. A language L is Strictly k-local on a Tier T iff there exists a tier $T \subseteq \Sigma$ and finite set $\mathrm{S} \subseteq \mathrm{F}_k\ (\rtimes T^* \ltimes)$ such that $L = \{w \in \Sigma^* : F_k(\rtimes E_T(w)\ltimes) \subseteq S\}$. Where S represents the permissible k-factors on the tier T, and elements in $F_k(\rtimes E_T(w)\ltimes) - S$ represent the forbidden k-factors on tier T. A language L is a Tier-based Strictly Local iff it is Stric4tly k-Local on Tier T for some $T \subseteq \Sigma$ and $k \in \mathrm{N}$. (Heinz et al., 2011: 60)

A.4 Input Output TSL (IO-TSL)

The power of TSL can be increased by changing the nature of the tier projection π, and making it an input-output strictly local transduction. The IO-TSL class is formally defined below (definitions and examples taken from Graf and Mayer (2018)).

The tier projection mechanism of IO-TSL is defined in terms of contexts that specify when a given symbol should be added to the tier. An (i, j)*-context* c is a 4-tuple $\langle \sigma, b, a, t\rangle$ with $\sigma \in \Sigma$, t a string over $\Sigma \cup \{\rtimes\}$ of length $j-1$, and a and b strings over $\Sigma \cup \{\rtimes, \ltimes\}$ of combined length $i-1$. The basic idea is that c specifies that σ should be projected whenever both of the following hold: it occurs between the substrings b (look-*b*ack) and a (look-*a*head), and the tier constructed so far ends in t. Given a set of contexts $c_1, c_2, \ldots, c_n$, we call it an (i, j)*-context set* iff for every c_m ($1 \leq m \leq n$) there are $i_m \leq i$ and $j_m \leq j$ such that c_m is an (i_m, j_m)-context.

Definition 9. Let C be an (i, j)-context set. Then the *input-output strictly* (i, j)*-local* (IOSL-(i, j)) tier projection π_C maps every $s \in \Sigma^*$ to $\pi'_C(\rtimes^i, s\ltimes^i, \rtimes^j)$, where for $\sigma \in \Sigma$ and $a, b, t, u, v, w \in (\Sigma \cup \{\rtimes, \ltimes\})^*$, it holds that $\pi'_C(ub, \sigma av, wt)$ is

$$\begin{array}{ll} \varepsilon & \text{if } \sigma av = \varepsilon, \\ \sigma\pi'_C(ub\sigma, av, wt\sigma) & \text{if } \langle \sigma, b, a, t\rangle \in C, \\ \pi'_C(ub\sigma, av, wt) & \text{otherwise.} \end{array}$$

Example 21. Let $\Sigma := \{a, b, c\}$ and consider the tier projection that always projects the first and last symbol of the string, always projects a, never projects c, and projects b only if the previous symbol on the tier is a. This projection is IOSL-(2,2). The context set contains all the contexts below, and only those:

- $\langle \sigma, \rtimes, \varepsilon, \varepsilon\rangle$ for all $\sigma \in \Sigma$,

- $\langle \sigma, \varepsilon, \ltimes, \varepsilon \rangle$ for all $\sigma \in \Sigma$,
- $\langle a, \varepsilon, \varepsilon, \varepsilon \rangle$,
- $\langle b, \varepsilon, \varepsilon, a \rangle$.

Definition 10. A stringset $L \subseteq \Sigma^*$ is *input-output tier-based strictly* (i, j, k)*-local* (IO-TSL-(i, j, k)) iff there exists an IOSL-(i, j) tier projection π_C and an SL-k language K such that $L := \{s \in \Sigma^* \mid \pi_C(s) \in K\}$. It is IO-TSL iff it is IO-TSL-(i, j, k) for some i, j, and k.

A.5 Interval-Based Strictly Piecewise (IBSP)

The definitions below are taken from Shafiei and Graf (2020).

Definition 11 (k-val)**.** A *segmented k-interval* ($k \geq 0$) over alphabet Σ, or simply *segmented k-val*, is a tuple $\langle L, R, F_i \rangle_{0 \leq i \leq k}$ such that

- $L, R \subseteq \Sigma \cup \{\varepsilon\}$ specify the *left edge* and *right edge*, respectively, and
- $F_i \subseteq \Sigma$ specifies the i-th *filler* slot.

Definition 12 (IBSP-k)**.** Let Σ be some fixed alphabet and $\rtimes, \ltimes \notin \Sigma$ two distinguished symbols. An IBSP-k grammar over Σ ($k \geq 0$) is a pair $G := \langle i, S \rangle$, where i is a segmented k-val over $\Sigma \cup \{\rtimes, \ltimes\}$ and $S \subseteq (\Sigma \cup \{\rtimes, \ltimes\})^k$ is a set of forbidden k-grams. A string $s \in \Sigma^*$ is generated by G iff there is no k-gram $u_1 \cdots u_k \in S$ such that $\rtimes^k s \ltimes^k$ is a member of the language

$$(\Sigma \cup \{\rtimes, \ltimes\})^* \times L \times F_0^* \times \{u_1\} \times F_1^* \times \{u_2\} \times \cdots \times F_{k-1}^* \times \{u_k\} \times F_k^* \times R \times (\Sigma \cup \{\rtimes, \ltimes\})^*$$

The language $L(G)$ is the set of all $s \in \Sigma^*$ that are generated by G. A stringset L is IBSP-k iff $L = L(G)$ for some IBSP-k grammar G.

Bibliography

Abdollahnejad, Elias and Dennis Storoshenko. 2020. Syntactic and semantic constraints on pronoun and anaphor resolution in persian. In *Advances in iranian linguistics*, chapter 2. John Benjamins Publishing Company. To appear.

Adger, David. 2003. *Core syntax: A minimalist approach*, volume 20. Oxford University Press Oxford.

Aksënova, Alëna. 2020. Tool-assisted induction of subregular languages and mappings. Doctoral dissertation, State University of New York at Stony Brook.

Aksënova, Alëna, Thomas Graf, and Sedigheh Moradi. 2016. Morphotactics as tier-based strictly local dependencies. In *Proceedings of the 14th SIGMORPHON workshop on computational research in phonetics, phonology, and morphology*, 121–130. Berlin, Germany: Association for Computational Linguistics.

Antonenko, Andrei. 2012. Feature-based binding and phase theory. Doctoral dissertation, The Graduate School, Stony Brook University: Stony Brook, NY.

Aoun, Joseph and Dominique Sportiche. 1983. On the formal theory of government. *Linguistic Review* 2:211–236.

Applegate, Richard. 1972. Ineseño chumash grammar. Doctoral dissertation, University of California, Berkeley.

Baek, Hyunah. 2017. Computational representation of unbounded stress: Tiers with structural features. *Proceedings of CLS* 53:13–24.

Belletti, Adriana and Luigi Rizzi. 1988. Psych-verbs and θ-theory. *Natural Language & Linguistic Theory* 291–352.

Boeckx, Cedric. 2012. *Syntactic islands*. Cambridge University Press.

Bošković, Željko. 2005. Left branch extraction, structure of np, and scrambling. *The free* *word order phenomenon: Its syntactic sources and diversity* 13–73.

Bošković, Željko. 2016. On the timing of labeling: Deducing comp-trace effects, the subject condition, the adjunct condition, and tucking in from labeling. *The Linguistic Review* 33:17–66.

Bresnan, Joan. 1977. Variables in the theory of transformations. In *Formal syntax*, eds. A. Akmajian, P. Culicover, and T. Wasow, 157–196. Academic Press.

Brody, Michael. 2015. One-dimensional syntax. *Manuscript. http://ling. auf. net/lingbuzz/002863* .

Brody, Michael. 2018. Two advantages of precedence syntax. In *Boundaries crossed, at the interfaces of morphosyntax, phonology, pragmatics and semantics*, 319–326. Springer.

Bruening, Benjamin. 2014. Precede-and-command revisited. *Language* 342–388.

Büring, Daniel. 2005. *Binding theory*. Cambridge University Press.

Champollion, Lucas. 2008. Binding theory in LTAG. In *Proceedings of the 9th international workshop on tree adjoining grammars and related formalisms*, eds. Claire Gardent and Anoop Sarkar, 1–8.

Chandlee, Jane. 2014. Strictly local phonological processes. Doctoral dissertation, University of Delaware.

Chandlee, Jane. 2017. Computational locality in morphological maps. *Morphology* 27:599–641.

Chao, Wynn. 1981. Pro-drop languages and nonobligatory control. *University of Massachusetts Occasional Papers in Linguistics* 7:46–74.

Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on information theory* 2:113–124.

Chomsky, Noam. 1980. On binding. *Linguistic inquiry* 11:1–46.

Chomsky, Noam. 1981. Lectures on government and binding, foris, dordrecht. *ChomskyLectures on Government and Binding1981* .

Chomsky, Noam. 1986a. *Barriers*. Cambridge, MA: MIT Press.

Chomsky, Noam. 1986b. *Knowledge of language: Its nature, origin, and use*. Greenwood

Publishing Group.

Chomsky, Noam. 1993. *Lectures on government and binding: The pisa lectures*. 9. Walter de Gruyter.

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

Chomsky, Noam. 2000. Minimalist inquiries: The framework. *Step by step: Essays on minimalist syntax in honor of Howard Lasnik* 89–155.

Chomsky, Noam. 2001. Derivation by phase. In *Ken hale: A life in language*, ed. Michael Kenstowicz, 1–52. Cambridge, MA: MIT Press.

Chomsky, Noam and Howard Lasnik. 1977. Filters and control. *Linguistic inquiry* 8:425–504.

Chomsky, Noam and Howard Lasnik. 1993. Principles and parameters theory. *Syntax: an International Handbook of Contemporary Research. Berlin: Walter de Gruyter* .

Cinque, Guglielmo. 1990. *Types of Ā-dependencies*. MIT press Cambridge, MA.

Corver, Norbert Ferdinand Marie. 1991. The syntax of left branch extractions. Doctoral dissertation, Katholieke Universiteit Brabant.

De Santo, Aniello and Thomas Graf. 2017. Structure sensitive tier projection: Applications and formal properties. Ms., Stony Brook University.

De Santo, Aniello and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In *International conference on formal grammar*, 35–50. Springer.

Diesing, Molly. 1992. *Indefinites*. MIT press.

Dinnsen, Daniel A. 1985. A re-examination of phonological neutralization. *Journal of linguistics* 21:265–279.

Elmedlaoui, Mohamed. 1995. Aspects des représentations phonologiques dans certaines langues chamito-sémitiques. Doctoral dissertation, Université Mohammed V Faculté des lettres et des siences humaines.

Epstein, Samuel David, Erich M Groat, Ruriko Kawashima, and Hisatsugu Kitahara. 1998. *A derivational approach to syntactic relations*. Oxford University Press.

Everaert, Martin. 1986. *The syntax of reflexivization*. Dordrecht: Foris.

Everaert, Martin. 1990. Nominative anaphors in icelandic: morphology or syntax? In *Issues in germanic syntax*, 277–306. De Gruyter Mouton.

Frank, Robert and K Vijay-Shanker. 2001. Primitive c-command. *Syntax* 4:164–204.

Frey, Werner and Hans-Martin Gärtner. 2002. On the treatment of scrambling and adjunction in Minimalist grammars. In *Proceedings of the conference on Formal Grammar*, 41–52.

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

García, Pedro and José Ruiz. 2004. Learning k-testable and k-piecewise testable languages from positive data. *Grammars* 7:125–140.

Gold, E Mark. 1967. Language identification in the limit. *Information and control* 10:447–474.

Graf, Thomas. 2012a. Locality and the complexity of minimalist derivation tree languages. In *Formal grammar*, 208–227. Springer.

Graf, Thomas. 2012b. Movement-generalized Minimalist grammars. In *LACL 2012*, eds. Denis Béchet and Alexander J. Dikovsky, volume 7351 of *Lecture Notes in Computer Science*, 58–73.

Graf, Thomas. 2013a. Local and transderivational constraints in syntax and semantics. Doctoral dissertation, UCLA.

Graf, Thomas. 2013b. The syntactic algebra of adjuncts. In *Proceedings of CLS 49*. To appear.

Graf, Thomas. 2017. The power of locality domains in phonology. *Phonology* 34:385–405.

Graf, Thomas. 2018a. Locality domains and phonological c-command over strings. In *Proceedings of NELS 2017*.

Graf, Thomas. 2018b. Why movement comes for free once you have adjunction. In *Proceedings of CLS 53*, 117–137.

Graf, Thomas, Alëna Aksënova, Hyunah Baek, Aniello De Santo, Hossep Dolatian, Sedigheh Moradi, Jon Rawski, Suji Yang, and Jeffrey Heinz. 2018. Tiers and relativized locality across language modules. Slides of a talk given at the 1-day workshop Parallels Between Phonology and Syntax, July 9, Meertens Instituut, Amsterdam, Netherlands.

Graf, Thomas and Jeffrey Heinz. 2015. Commonality in disparity: The computational view of syntax and phonology. *Slides of a talk given at GLOW* 18.

Graf, Thomas and Connor Mayer. 2018. Sanskrit n-retroflexion is input-output tier-based strictly local. In *Proceedings of the fifteenth workshop on computational research in phonetics, phonology, and morphology*, 151–160.

Graf, Thomas and Nazila Shafiei. 2019. C-command dependencies as TSL string constraints. In *Proccedings of SCiL 2019*, 205–215.

Haegeman, Liliane. 1994. *Introduction to government and binding theory*. Wiley-Blackwell.

Hansson, Gunnar Ólafur. 2010. Long-distance voicing assimilation in berber: Spreading and/or agreement. In *Proceedings of the 2010 annual conference of the canadian linguistic association*.

Harbert, Wayne. 1995. Binding theory, control, and pro. *Government and binding theory and the minimalist program* 177–240.

Hasegawa, Hiroshi. 2005. Reflexive binding as agreement and its interactions with the phase system. *The world of linguistic research: A festschrift for Kinsuke Hasegawa on the occasion of his seventieth Birthday, ed. Noriko Imanishi* 53–69.

Hayes, Bruce. 1995. *Metrical stress theory: Principles and case studies*. University of Chicago Press.

Heinz, Jeffrey. 2010. Learning long-distance phonotactics. *Linguistic Inquiry* 41:623–661.

Heinz, Jeffrey. 2018. The computational nature of phonological generalizations. In *Phonological typology*, Phonetics and Phonology, chapter 5, 126–195. Mouton De Gruter.

Heinz, Jeffrey, Colin de la Higuera, and Menno van Zaanen. 2015. *Grammatical inference for computational linguistics*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Heinz, Jeffrey, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th annual meeting of the association for computational linguistics*, 58–64.

Heinz, Jeffrey and James Rogers. 2013. Learning subregular classes of languages with factored deterministic automata. In *Proceedings of the 13th meeting on the mathematics of language (mol 13)*, 64–71.

Hicks, Glyn. 2009. *The derivation of anaphoric relations*, volume 139. John Benjamins Publishing.

Higginbotham, James. 1985. On semantics. *Linguistic inquiry* 547–593.

Hornstein, Norbert, Jairo Nunes, and Kleanthes K Grohmann. 2005. *Understanding minimalism*. Cambridge University Press.

Huang, C-T James. 1998. Logical relations in chinese and the theory of grammar. Doctoral dissertation, MIT.

Hyman, Larry M. 2011. Tone: Is it different. *The handbook of phonological theory* 2:197–239.

Jäger, Gerhard and James Rogers. 2012. Formal language theory: refining the chomsky hierarchy. *Philosophical Transactions of the Royal Society B: Biological Sciences* 367:1956–1970.

Jardine, Adam and Jeffrey Heinz. 2016. Learning tier-based strictly 2-local languages. *Transactions of the Association for Computational Linguistics* 4:87–98.

Jensen, John T. 1974. A constraint on variables in phonology. *Language* 675–686.

Johnson, C Douglas. 1972. *Formal aspects of phonological description*. Mouton, The Hague.

Kandybowicz, Jason. 2006. Comp-trace effects explained away. In *Proceedings of the 25th west coast conference on formal linguistics*, volume 220, 228. Citeseer.

Kaplan, Abby. 2010. Phonology shaped by phonetics: The case of intervocalic lenition. Doctoral dissertation, University of California, Santa Cruz.

Kaplan, Ronald M. and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378.

Karimi, Simin. 2005. *A minimalist approach to scrambling: Evidence from persian*, volume 76. Walter de Gruyter.

Karttunen, Lauri, Ronald M Kaplan, and Annie Zaenen. 1992. Two-level morphology with composition. In *Coling 1992 volume 1: The 15th international conference on computational linguistics*.

Kawamura, Tomoko. 2004. A feature-checking analysis of japanese scrambling. *Journal* *of Linguistics* 40:45–68.

Kayne, Richard. 2002. Pronouns and their antecedents. *Derivation and explanation in the minimalist program* 133:166.

Kayne, Richard S. 1994. *The antisymmetry of syntax*, volume 25. MIT press.

Kiparsky, Paul. 2002. Disjoint reference and the typology of pronouns. In *More than words*, eds. Ingrid Kaufmann and Barbara Stiebels, volume 53 of *Studia Grammatica*, 179–226. Berlin: Akademie Verlag.

Kobele, Gregory M., Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to Minimalism. In *Model theoretic syntax at 10*, eds. James Rogers and Stephan Kepser, 71–80.

Kornai, Andras. 1985. Natural languages and the chomsky hierarchy. In *Second conference of the european chapter of the association for computational linguistics*.

Larson, Richard K. 1988. On the double object construction. *Linguistic inquiry* 19:335–391.

Larson, Richard K. 2009. *Grammar as science*. MIT Press.

Laszakovits, Sabine and Thomas Graf. 2020. Diagnosing movement via the absence of c-command relations. *University of Pennsylvania Working Papers in Linguistics* 26:16.

Lebeaux, David. 1991. Relative clauses, licensing, and the nature of the derivation. In *Perspectives on phrase structure: Heads and licensing*, 209–239. Brill.

Lee-Schoenfeld, Vera. 2004. Binding by phase:(non-) complementarity in german. *Journal of Germanic Linguistics* 16:111–171.

Manzini, M Rita and Kenneth Wexler. 1987. Parameters, binding theory, and learnability. *Linguistic inquiry* 413–444.

Mayer, Connor. 2021. Capturing gradience in long-distance phonology using probabilistic tier-based strictly local grammars. *Proceedings of the Society for Computation in Linguistics* 4:39–50.

Mayer, Connor and Travis Major. 2018. A challenge for tier-based strict locality from Uyghur backness harmony. In *International conference on formal grammar*, 62–83. Springer.

McMullin, Kevin James. 2016. Tier-based locality in long-distance phonotactics: learnability and typology. Doctoral dissertation, University of British Columbia.

Michaelis, Jens. 2001. Transforming linear context-free rewriting systems into Minimalist grammars. *Lecture Notes in Artificial Intelligence* 2099:228–244.

Miyagawa, Shigeru. 2004. The nature of weak islands. *Ms., Massachusetts Institute of Technology, Cambridge* .

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

Odden, David. 1994. Adjacency parameters in phonology. *Language* 289–330.

Pasternak, Robert and Thomas Graf. 2021. Cyclic scope and processing difficulty in a minimalist parser. *Glossa: a journal of general linguistics* 6.

Pesetsky, David and Esther Torrego. 2007. The syntax of valuation and the interpretability of features. *Phrasal and clausal architecture: Syntactic derivation and interpretation* 262–294.

Pica, Pierre. 1987. On the nature of the reflexivization cycle. In *Proceedings of NELS17*. UMass, Amherst.

Postal, Paul Martin. 1998. *Three investigations of extraction*. MIT Press.

Rappaport, Gilbert C. 1986. On anaphor binding in russian. *Natural Language & Linguistic Theory* 4:97–120.

Reinhart, Tanya and Eric Reuland. 1993. Reflexivity. *Linguistic inquiry* 24:657–720.

Reinhart, Tanya Miriam. 1976. The syntactic domain of anaphora. Doctoral dissertation, Massachusetts Institute of Technology.

Reuland, Eric. 2006. Agreeing to bind. *Festschrift for Henk van Rimsdijk* .

Reuland, Eric and Jan Koster. 1991. Long distance anaphora: an overview. In *Long distance anaphora*, chapter 1, 1–26. Cambridge University Press.

Ritter, Elisabeth and Anna Szabolcsi. 1985. Let's take the that-trace effect seriously. ms.

Rizzi, Luigi. 1990. *Relativized minimality.* The MIT Press.

Rizzi, Luigi. 2001. Relativized minimality effects. *The handbook of contemporary syntactic* *theory* 4:89–110.

Rizzi, Luigi. 2004. Locality and left periphery. *Structures and beyond: The cartography of syntactic structures* 3:223–251.

Rizzi, Luigi. 2007. On some properties of criterial freezing. *Studies in linguistics* 1:145–158.

Rizzi, Luigi and Ur Shlonsky. 2007. Strategies of subject extraction. *Interfaces+ recursion= language? Chomsky's minimalism and the view from syntax-semantics* 115–160.

Rogers, James. 1998. *A descriptive approach to language-theoretic complexity*. Citeseer.

Rogers, James, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Visscher, David Wellcome, and Sean Wibel. 2009. On languages piecewise testable in the strict sense. In *The mathematics of language*, 255–265. Springer.

Ross, John R. 1967. Constraints on variables in syntax. Doctoral dissertation, MIT.

Sabel, Joachim. 2002. A minimalist analysis of syntactic islands. *Linguistic Review* 19:271–315.

Shafiei, Nazila and Thomas Graf. 2020. The subregular complexity of syntactic islands. *Proceedings of the Society for Computation in Linguistics* 3:272–281.

Shieber, Stuart M. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8:333–345.

Sprouse, Jon, Matt Wagers, and Colin Phillips. 2012. A test of the relation between working-memory capacity and syntactic island effects. *Language* 82–123.

Stabler, Edward P. 1997. Derivational Minimalism. In *Logical aspects of computational linguistics*, ed. Christian Retoré, volume 1328 of *Lecture Notes in Computer Science*, 68–95. Berlin: Springer.

Stabler, Edward P. 2011. Computational perspectives on Minimalism. In *Oxford handbook of linguistic Minimalism*, ed. Cedric Boeckx, 617–643. Oxford: Oxford University Press.

Starke, Michal. 2001. Move dissolves into merge: a theory of locality. Doctoral dissertation, Université de Genève.

Stepanov, Arthur. 2007. The end of CED? minimalism and extraction domains. *Syntax* 10:80–126.

Storoshenko, Dennis. 2017. Person asymmetries in english controlled and exempt anaphors. Slides of a talk given at the Manitoba Workshop on Person, September 22-23, Winnipeg, Manitoba.

Szabolcsi, Anna. 2006. Strong vs. weak islands. *The Blackwell Companion to Syntax* 479–531.

Takahashi, Daiko. 1994. Minimality of movement. Doctoral dissertation, University of Connecticut.

Tang, Chih-Chen Jane. 1989. Chinese reflexives. *Natural Language & Linguistic Theory* 7:93–121.

Thráinsson, Höskuldur. 1991. Long-distance reflexives and the typology of NPs. *Long-distance anaphora* 49–75.

Thráinsson, Höskuldur. 2017. Icelandic long distance reflexives revisited. Slides of a talk given at the 48th meeting of the North East Linguistic Society.

Truswell, Robert. 2007. Extraction from adjuncts and the structure of events. *Lingua* 117:1355–1377.

Vennemann, Theo. 1968. German phonology. Doctoral dissertation, University of California, Los Angeles.

Vu, Mai Ha. 2018. Towards a formal description of NPI-licensing patterns. *Proceedings of the Society for Computation in Linguistics* 1:154–163.

Vu, Mai Ha. 2020. A quantifier-based approach to npi-licensing typology: Empirical and computational investigations. Doctoral dissertation, University of Delaware.

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* 2:267–276.

Yang, Su Ji. 2018. Subregular complexity in Korean phonotactics. Undergraduate honors thesis, Stony Brook University.

Zwart, Jan-Wouter. 2002. Issues relating to a derivational theory of binding. *Derivation and explanation in the minimalist program* 269–304.