UNIVERSITY OF CALIFORNIA
Los Angeles

Logics of Phonological Reasoning

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Arts in Linguistics

by

Thomas Graf

2010

The thesis of Thomas Graf is approved.

Robert Daland

Edward L. Keenan

Kie Zuraw

Edward P. Stabler, Committee Chair

University of California, Los Angeles

2010

To: whoever knows their name belongs here

Table of Contents

List of Figures

List of Tables

Acknowledgments

A short comparison between this thesis and earlier publications of mine on the topics discussed herein (Graf 2010a,b) makes it patently obvious that I benefited a lot from the input of my committee members. I would especially like to thank Ed Stabler and Kie Zuraw; while this project was still in its infancy, both of them suggested to take it in a more empirical direction, which resulted in what might prove to be the most important contributions of this undertaking. Kie was also of great help in my subsequent search for empirical phenomena instantiating the relevant formal patterns, and so was Bruce Hayes. I am also indebted to Craig Melchert for answering all my emails concerning Sanskrit, and doing so at super-human speed to boot. The same can be said about Marcus Kracht and my emails concerning technical details of mathematical model theory. Finally, it would be shameful for me not to pay off my debts of the past by failing to mention how much I learned from the Vienna phonology group, in particular Marcus Pöchtrager. The answer: almost everything I know about phonology (*modulo* OT, of course ;-)).

During its transmogrification, various parts of the thesis were presented at the 33rd Penn Linguistics Colloquium and the ESSLLI 2009 Student Session, and the comments and questions of the respective reviewers and/or audiences led to numerous improvements in the exposition. It should come as no surprise to the reader that the usual *mea culpa* applies.

ABSTRACT OF THE THESIS

Logics of Phonological Reasoning

by

Thomas Graf

Master of Arts in Linguistics

University of California, Los Angeles, 2010

Professor Edward P. Stabler, Chair

Inspired by Kracht (Kracht 2003) and Potts and Pullum (Potts and Pullum 2002), who use tools from mathematical logic in their investigation of phonological theories, I develop an extendable modal logic over string structures, which in turn is used to formalize a specific phonological theory, Government Phonology. Building on this logical foundation, I compare Government Phonology to SPE and arrive at the surprising result that Government Phonology corresponds to a very weak fragment of SPE yet can attain the latter's full expressivity by extending the power of feature spreading. I then exhibit two attested phonological phenomena that require moving beyond the power of standard Government Phonology: n-retroflexion in Sanskrit and primary stress assignment in Creek and Cairene Arabic. I further identify several other axes along which Government Phonology can be generalized, moving us towards a parametric metatheory of phonology.

These results are of interest to linguists as they (i) establish a lower bound on the power every descriptively adequate phonological theory has to make available, (ii) tell us how this power can be measured in a precise way, and (iii) show how different phonological proposals are related to each other, thus complementing the well-established empirical methods of theory comparison. Computational

phonologists, on the other hand, will appreciate the explicit formalization of a subregular theory of phonology and the (implicit) demonstration that the majority of empirical phenomena does not need full finite-state expressivity.

Most of the results reported herein can also be found in Graf (2010a,b). This thesis, however, vastly exceeds them in terms of clarity, exhaustiveness and preciseness, thanks to the lack of page restrictions.

Introduction

One often finds that scrutinizing the world around you — already instructive by itself — comes with the added bonus of also teaching you something new about yourself. From this perspective, it is easy to appreciate the prominent position that comparisons of different frameworks have always enjoyed in modern linguistics. Some might deride it as a trademark of any field that is still in its infancy, with a lot of competing approaches, methodologies and ontologies trying to establish themselves as the predominant paradigm. Be that as it may, the merits of reflecting on the properties of one's theory of choice and how it relates to the rest of the field cannot be disputed. For instance, Michael Brody's defense of his mirror theory has furthered our understanding of the differences between representational and derivational frameworks (see e.g. Brody 1995, 2002), and Johnson and Lappin (1997) and Sternefeld (1996) were among the first to seriously investigate the role of optimality conditions in syntax.

Strikingly, though, all these comparative studies are ultimately based on empirical considerations, that is, their results are obtained by meticulous analysis of empirical data. Moreover, the few studies that rely on purely formal reasoning (Kornai and Pullum 1990, for example) have mostly been ignored by the linguistic community. The importance of conceptual arguments such as minimality considerations has admittedly increased considerably in recent years in the wake of Chomsky's Minimalist Program, but the general consensus still seems to be that such reasoning by itself does not tarnish a proposal, let alone refute it.

In this thesis, I argue for a new, formally grounded approach to comparing theories in the realm of phonology: model-theoretic phonology (MTP). MTP is based on the insight that many attributes of a theory are reflected in the properties of the

weakest language one can use to describe it. In model-theoretic approaches, this description language is some logic chosen from the array of logics one encounters in mathematics and computer science. In particular, a set of logical formulas is a formalization of a linguistic theory if the structures that do not violate any of these formulas are exactly those that are deemed grammatical by the theory. Since mathematical logic is a field with rich connections to formal language theory and complexity theory, logical formalization can act as a link between linguistics and those fields. These bonds also allow it to unearth the implicit cognitive claims of linguistic proposals, thus making theoretical linguistics (at least partially) amenable to psycholinguistic research and allowing it to break out of its "competence sandbox" (see Pullum and Rogers 2006 for further elaboration of this point).

MTP complements the established empirical methods (rather than obsoleting them) in that for many problems it involves considerably less analytical toiling and permits us to rigorously prove claims that the latter cannot even formulate in its accustomed vocabulary. It does all that while at the same time remaining sufficiently transparent so that linguists can easily evaluate the claims that are put forward. The reader will have plenty of opportunities to convince himself of the practicality of the approach when I put it to good use in order to compare a particular theory, Government Phonology, to SPE, the shortcomings of which it set out to rectify. In particular, he or she will see that the results tell us as much about Government Phonology itself as about its relation to SPE, and that they translate into interesting empirical questions to boot. Among other things, I establish that:

- Government Phonology is strictly weaker than SPE.
- Modifying a single parameter of Government Phonology, one can subsequently increase its power until it reaches the level of SPE.

- Phenomena that invariably require moving beyond the power of Government Phonology are scarce.

It should be pointed out that logical approaches have already been used to great effect in the realm of syntax for two decades. Among other things, it was shown that late Government-and-Binding theory in the spirit of Rizzi (1990) is weaker than Minimalist syntax (a corollary of the results in Rogers 1998 and Michaelis 2001) and that the kind of universals envisioned by the Principles-and-Parameters approach are fundamentally different from universals in GPSG (Rogers 1996) (see Pullum 2007 for a survey with further results). By putting the focus on the structures licensed by a linguistic theory, it also becomes easier to see how this theory would have to be tweaked to account for phenomena it is currently too weak for.

Hopefully this handful of examples will have the reader convinced that my interests are of a genuinely linguistic nature, and that the method I use can benefit them beyond the immediate results on Government Phonology reported herein. This is not to say that only phonologists will find something of interest in here. Computational linguists might appreciate the formalization of a rather atypical phonological theory as well as the investigation of the generative capacity of phonology in the last chapter — while we have a good mathematical understanding of the power of syntax nowadays, the same does not hold true of phonology, mostly because it is presumably weaker than any of the formal language classes usually studied in computer science. Even logicians could find something of interest here, as some of the questions I can only briefly touch upon translate into challenging problems in finite model-theory.

The thesis is laid out in three chapters that are dedicated to the linguistic preliminaries, the logical formalization and the comparison of Government Phonology and SPE, respectively. In the first chapter, I start out with a very brief overview of

SPE, highlighting some of the more obscure parts of the machinery that are easily forgotten. More importance, though, is put on the mathematical properties of SPE, which most readers are likely not familiar with. After this short excursus into formal terrain, I swiftly return to purely linguistic matters in my extensive introduction to Government Phonology. Since this framework also marks new terrain for many readers, I proceed rather slowly, relying throughout the presentation on examples and figures, but not to the detriment of precise definitions. My goal is to convey to the reader just how different Government Phonology and SPE are (at least at first sight), yet at the same time provide enough of a working knowledge so that one can follow the logical formalization in Chap. 2. There I first engage in a detailed discussion of MTP and its general methodology, focussing in particular on its advantages over empirical as well as competing formal approaches. This is followed by a beginner-friendly introduction to mathematical logic, before I finally turn to the formalization itself. In the last chapter, I show how we can map logics and the phonological theories they formalize onto a phonological expressivity hierarchy, and I exhibit two empirical phenomena, Sanskrit [n]-retroflexion and primary stress assignment in Cairene Arabic and Creek, that exceed the power of Government Phonology, but not of SPE. I close with a thorough evaluation of the results, carefully contextualizing them and pointing out their limitations so that they won't be misconstrued.

The thesis contains two appendices. The first one explains basic mathematical terminology that I do not define anywhere else; it should be unnecessary for anybody who has taken at least an introduction to semantics or philosophical logic at the undergraduate level. The second appendix is devoted to a proof of Thm. 3.2 that uses Ehrenfeucht-Fraïssé pebble games. It is not for the faint of heart.

CHAPTER 1

SPE & Government Phonology

1.1 Overview

This chapter serves a dual purpose. On the one hand, it provides an introduction to SPE and Government Phonology (with the former proceeding at a considerably brisker pace than the latter) and thus lays the foundation for the more technical discussion that ensues in the chapters 2 and 3. On the other hand, it is meant to give the reader a feeling for how different the two theories are so that the success of MTP in positioning GP with respect to SPE can be fully appreciated. In Sec. 1.2 I start out with a very short reminder of the formalism itself before turning my attention to its treatment in the computational literature. This is followed up by a (hopefully) accessible introduction to Government Phonology in Sec. 1.3.

1.2 SPE

Since it is safe to assume that the reader is familiar with at least the basics of SPE as put forward in Chomsky and Halle (1968), my primary focus will be on the mathematical properties of SPE.

Just like transformational syntax before the introduction of Government-and-Binding Theory in Chomsky (1981) (i.e. Chomsky 1957, Chomsky 1965), SPE is a derivational formalism in which rewriting rules apply to *underlying representations*

to yield surface representations. An underlying representation is a finite string of sounds represented by matrices over finitely-valued features drawn from a finite set $\mathscr{F}$ of features (in the case of SPE, $\mathscr{F}$ contains between 20 and 24 features, most of which are binary-valued and defined in terms of articulatory phonetics). The strings may also contain diacritic symbols such as # and + for word and morpheme boundaries, respectively, which for the sake of uniformity can also be encoded by matrices.

The matrices are then manipulated by a sequence of rewrite rules of the form $R := \alpha \rightarrow \beta \mid \gamma \_\_ \delta$, where α is a matrix or the empty string and β , γ, δ are either strings of matrices or the empty string. A rule like R encodes the instruction that the substring $\gamma\alpha\delta$ is to be rewritten as $\gamma\beta\delta$. A special case are transformational rules which switch the position of adjacent segments. They can be viewed as one of many notational devices that serve to condense multiple rules into one, e.g. various kind of brackets and variables for feature values. It is important to realize, though, that those enriched rules aren't proper rewrite rules but rather rule schemata that have to be compiled out to obtain the intended rewrite rules.[1] Rules can be optional or mandatory, and some can be unordered relative to each other with respect to the timing of their application. Moreover, the directionality of rules is also parameterized, with some applying from left-to-right, some from right-to-left, and some simultaneously to all matching substrings. Consider the rule $a \rightarrow b \mid ab \_\_ ba$ that turns $ababa$ into $abbba$. Given the input string $abababa$, left-to-right application of the rule yields the string $abbbaba$, whereas right-to-left application yields $ababbba$. The output of simultaneous application is $abbbbba$.

[1] Nonetheless they have an important role to play in SPE regarding *naturalness*: the naturalness of a process is measured by the complexity of the shortest rule schema that encodes it. Moreover, disjunctive rule ordering, i.e. cases where rule R may be applied only if the adjacent rule R' cannot, is permissible only if the two rules can be collapsed into a schema by using bracket notation to indicate optionality of segments.

The way rules are allowed to be applied iteratively is of utmost importance in determining what kind of languages can be generated in SPE. Formal language theory tells us that the rules of SPE are the rules of a context-sensitive grammar, so in terms of the Chomsky hierarchy of formal languages, SPE should be capable of deriving at least context-sensitive languages (also known as Type-1 languages), which are significantly more powerful than what is considered necessary even for syntax (see Fig. 1.1 on the following page). To add insult to injury, any SPE grammar in which the input of a rule may be rewritten by the empty string is in fact an unrestricted (also known as Type-0) rewriting system, i.e. SPE has the same power as a Turing machine. That is to say, if SPE cannot derive a language, no algorithmic computing device can. Most interestingly, though, Johnson (1972) observed that a simple restriction will ensure that SPE generates only regular languages, which are the weakest class in the Chomsky hierarchy: no rule may apply to its own output. Here "output" does not refer to the entire output string but only the material that was newly introduced by the rule or is the result of rewriting a symbol in the input string. Hence every rule may rewrite every symbol in the input at most once.

Consider the rule $a \rightarrow b \mid b\_\_$. Assume it applies from left-to-right, and that the input is baa. The first substring that is in the domain of the rule is ba, which is rewritten as bb, yielding bba. Now the rule as it is stated cannnot apply to the substring bb, but it can still apply to the substring ba, even though the b was created by a previous application of the rule. The important thing is that the rule will only rewrite a, which was not meddled with by any previous applications of the rule.

In order to see why this ban against rules rewriting their own output is important, consider the rule $\varepsilon \rightarrow ab \mid a\_\_b$, which allows us to insert the sequence ab after an

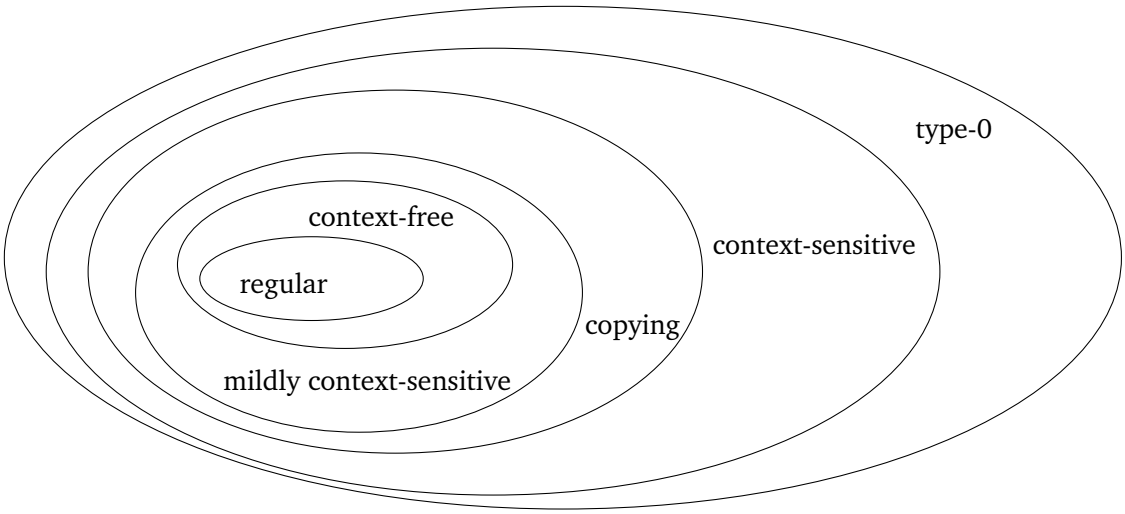


Figure 1.1: Extended version of the Chomsky hierarchy; both phonology and restricted SPE are regular, syntax is mildly context-sensitive, unbounded reduplication can be faithfully modelled only by copying languages, and standard SPE is type-0

a that is immediately followed by a b. When given the string ab ($= \varepsilon a \varepsilon b \varepsilon$), the rule will rewrite it as $aabb$ ($= \varepsilon a \varepsilon a \varepsilon b \varepsilon b \varepsilon$). With the ban in place, this is the end of the derivation. The reasoning goes as follows: the only possible target is the ε between the last a and the first b, but this is new material that was previously introduced by the rewriting rule in question (keep in mind that the entire substring $a\varepsilon b$ is new, the symbols of the input string are now at the edges of the output string, not in the middle). Since a rule may not apply to its own output, there are no licit targets left. Without the ban, however, we can continue the rewriting procedure, yielding $aaabbb$, $aaaabbbb$, and so on. The result will be the language $a^n b^n$, where n occurrences of a are followed by n occurrences of b, which is not a regular language (it is context-free).

The proof that the condition proposed by Johnson limits SPE to regular languages can be found in Kaplan and Kay (1994). It is very tedious, but the upshot is that all parameters of rule application (optionality, directionality, relative ordering) can be accommodated without problems. Kaplan and Kay even go so far as to

contend that all aspects of SPE that are invoked in linguistic analyses can be fit into the restricted variant of SPE. This claim should be taken with a grain of salt. To my knowledge, there are indeed no analyses that rely on rules rewriting their own output. However, there are arcane, seldom invoked parts of SPE that might be problematic. In his famous paper on conspiracies, for example, Kisseberth (1970) introduces rules that are triggered only if the input violates certain constraints. In Sommerstein (1974), these constraints are extended to full truth-conditions, which highlights the problem that if no restrictions are put on what qualifies as a licit constraint, we once again run risk of blowing the expressivity of SPE out of proportion.[2] A constraint like "rule *R* is triggered if and only if the string encodes the Gödel number of a theorem of first-order logic" could be used to generate languages that lie even outside the Chomsky hierarchy. That is to say, SPE would be capable of deriving non-computable languages. Thus a more appropriate interpretation of Kaplan and Kay's claim would view it as a conjecture about how much expressivity is likely necessary for phonology given what we know after decades of linguistic research (and, just as importantly, natural language processing). But this raises the question if phonology is indeed regular.

The consensus in the computational phonology community is that it is, except for one phenomenon: unbounded reduplication, in which strings of unbounded size are copied and then concatenated (see Albro 2005 for examples and a formal analysis). Copying is beyond the reach of what is commonly assumed to be sufficient even for syntax (but see Kobele 2006 for empirical evidence that copying is indispensable in syntax, too). So far, no fully satisfying answer has been given as to why phonology sticks with the power of regular languages yet at the same time seems to require means of expressivity that greatly exceed anything else found in

[2]My thanks go to Kie Zuraw for bringing this paper to my attention.

natural language. It is suspicious, though, that reduplication is a morphonological process, so its complexity might be an epiphenomenon of the intricate interaction of two considerably less complex individual systems, phonology and morphology. As I will restrict myself to phonology proper, it is probably safe to assume for our purposes that the power of regular languages is all one needs. In a certain way, then, SPE establishes an upper bound on how much expressivity is desirable for phonology.

1.3 Government Phonology

Returning to a strictly linguistic perspective on SPE, we may characterize it in broad strokes by the following attributes: it is derivational, assumes a very shallow and flat structure, and uses a plethora of binary features that are assembled into matrices. Government Phonology (GP) is its very opposite, as the reader will see in this section: decidedly constraint-based, highly structured, and restricted to a select few privative features which are again combined into structured expressions. *A priori*, it seems like the two formalisms have virtually nothing in common. The essence of the research reported here is that they are in fact very similar, but it takes a mathematically informed perspective to make the superficial differences disappear. In order to appreciate the result though, the reader has to be shown just how different GP is, which is the purpose of the remainder of this section.

First, though, a note on the sources used is in order. Just like Government-and-Binding theory, GP has changed a lot since its inception, and its practitioners hardly ever fully specify the details of their preferred version of GP. However, there seems to be a consensus that a GP-variant should be deemed canonical if it incorporates the following modules:

- government, the syllable template, coda licensing and the ECP from Kaye et al. (1990), and
- magic licensing from Kaye (1992), and
- licensing constraints and the revised theory of elements from Charette and Göksel (1996) and Kaye (2000, 2001).

The strategy I will pursue in the exposition is in line with this decree in so far as I follow the definitions in Kaye (2000) as closely as possible and fill in any gaps using the relevant literature.

1.3.1 Feature System

GP replaces SPE's system of binary or attribute valued features by a set of privative features called *elements*. Privative features take no values, they are either present on a segment or not; such features are commonly used in autosegmental analyses of tone systems (starting with Goldsmith 1976), where two privative features **L** and **H** — denoting low and high tone, respectively — are used to account for various tone patterns and their interaction. For instance, a toneless segment would lack both features, a high tone segment would be specified only for **H**, and a falling tone would be rendered as **H** spreading to the right into a position associated with **L**. GP's elements are a straight-forward generalization of such autosegmental tone features to all phonological features (impatient readers may go ahead and take a look at the examples in Tab. 1.4 on page 15).

In contrast to SPE, GP's feature system has undergone numerous revisions, all of which served the sole purpose of reducing the number of elements in order to bring the number of combinatorial possibilities in line with the typologically attested

maximum of roughly 120 phonemes[3] for any language. For the sake of simplicity, I discuss only the model described in Kaye (2000), which takes the set of elements to consist of **A**, **I**, **U**, **L**, **H** and ʔ; as the reader will see in later chapters, this has no ramifications for the validity of any formal claims I make about GP.

SPE's feature matrices are replaced by *phonological expressions* (PE), which consist of a possibly empty set of elements in *operator* position and at most one element in *head* position. No element is allowed to occur twice in a PE. By convention, PEs are written as pairs (in round brackets) with the first coordinate denoting the set of operators and the second the head. It is furthermore common practice to mark the head by underlining. However, if a position is empty, this is also indicated by an underline. Thus, given the set of elements proposed in Kaye (2000), the pairs $(\{\mathbf{A},\mathbf{I}\},\underline{\mathbf{U}})$, $(\{\mathbf{A}\},\_)$, and $(\{\_\},\_)$ are PEs. The pairs $(\{\mathbf{A}\},\{\underline{\mathbf{U}},\underline{\mathbf{I}}\})$ and $(\{\mathbf{A},\mathbf{I}\},\underline{\mathbf{A}})$, on the other hand, are not, because the second has more than one head, while the first contains two occurrences of the same element.

Regarding the mapping from PEs to sounds, beginners might follow Kaye et al. (1985) in treating elements as bundles of fully specified SPE features which are combined in a specific way to produce a SPE feature matrix. Under this conception, **A**, **I**, **U** and empty heads/operators are defined as in Tab. 1.1. The feature matrix of

$$\mathbf{A} := \begin{bmatrix} - & \text{lab} \\ + & \text{back} \\ - & \underline{\text{hi}} \\ - & \text{ATR} \\ + & \text{lo} \end{bmatrix} \qquad \mathbf{I} := \begin{bmatrix} - & \text{lab} \\ - & \underline{\text{back}} \\ + & \text{hi} \\ - & \text{ATR} \\ - & \text{lo} \end{bmatrix} \qquad \mathbf{U} := \begin{bmatrix} + & \underline{\text{lab}} \\ + & \text{back} \\ + & \text{hi} \\ - & \text{ATR} \\ - & \text{lo} \end{bmatrix} \qquad \_ := \begin{bmatrix} - & \text{lab} \\ + & \text{back} \\ + & \text{hi} \\ - & \text{ATR} \\ - & \text{lo} \end{bmatrix}$$

Table 1.1: The elements **A**, **I**, **U** as feature bundles

[3]Usage of the term phoneme is generally frowned upon by Government phonologists for various technical but plausible reasons. At rare occasions I will nevertheless make use of it, albeit in a purely descriptive way.

a PE is then calculated by first computing the matrix of the head and subsequently copying the values of any underlined features of the elements in operator position into the head's matrix. For the three PEs mentioned above, this procedure will yield the matrices in Tab. 1.2. As an easy exercise, the reader may try to compute the matrices for $(\{\_\}, \underline{\mathbf{A}})$ and $(\{\mathbf{A}\}, \underline{\mathbf{I}})$.

$$(\{\mathbf{A},\mathbf{I}\},\underline{\mathbf{U}}) = \begin{bmatrix} + & \text{lab} \\ - & \text{back} \\ - & \text{hi} \\ - & \text{ATR} \\ - & \text{lo} \end{bmatrix} \qquad (\{\mathbf{A}\},\_) = \begin{bmatrix} - & \text{lab} \\ + & \text{back} \\ - & \text{hi} \\ - & \text{ATR} \\ - & \text{lo} \end{bmatrix} \qquad (\{\_\},\_) = \begin{bmatrix} - & \text{lab} \\ + & \text{back} \\ + & \text{hi} \\ - & \text{ATR} \\ - & \text{lo} \end{bmatrix}$$

Table 1.2: Three phonological expressions of GP as SPE feature bundles

I refrain from presenting the matrices for the remaining elements, for the simple reason that while the matrix-based approach to elements might be more intuitive to phonologists accustomed to SPE-like feature systems, it misrepresents the current conception of elements by tying them to specific phonetic realizations. In contemporary GP, the phonetic interpretation of PEs is considered a matter of the phonology-phonetics interface rather than phonology proper, and crucially there might be minor language-specific differences with respect to the phonetic realization of a particular PE. The PE (_, _), for instance, yields a [ə] in German and a [ɨ] in Turkish. Therefore, the PE corresponding to some sound in some language can only be determined by investigation of its phonological behavior, i.e. which phonological well-formedness conditions it obeys and how it interacts with other sounds in this language. Furthermore, the actual contribution of elements differs between vowels and consonants and which position they occupy in a PE. The element **A**, for instance, may denote an [a]-like quality for vowels but coronality for consonants (again with some language-specific leeway). Clearly, such a loose

mapping from phonology to phonetics does not square well with a conception of elements as bundles of fully specified phonetic features.

Arguably the best way to think about elements at a phonetic level, then, is the following. For vowels, imagine a triangle representing the vowel system of the language. The head of the PE determines which corner to start at and the operators tell you in which direction you have to move. Thus ({**A**}, **I**) takes you from [i] towards [a], yielding [e], whereas ({**I**}, **A**) takes you from [a] towards [i], yielding [ɛ]. The same procedure applies for **U**, unless **A** and **I** are already present and **U** acts as an operator, in which case it will only contribute labiality. If the head or the operator are empty, take the tense or lax variant, respectively, of whatever sound is specified by the remainder of the PE. For example, ({_}, **I**) is mapped to [i], and ({**I**}, _) to [I]. Finally, the features **L** and **H** occur only as operators and add a low and a high tone, respectively, while ʔ never occurs in PEs for vowels at all.

Things are more complicated for consonants, mainly because most work on subsegmental structure that has been carried out in the GP tradition focuses on vowels. In particular, it is often difficult to decide for consonants which element should function as the head. With that said, the map from elements to phonetic properties is still fairly simple in most cases. In head position, **A**, **I** and **U** determine the place of articulation: alveolar, palatal and labial, respectively. If a consonant is velar, its head is empty. In operator position, these three elements add coronality, palatalization and velarization. The element ʔ is always used to denote stops, irrespective of its role in the PE. The remaining two elements **H** and **L** are usually restricted to operator position and may add a variety of properties. PEs for fricatives often (but not always) contain an **H**, which might also indicate aspiration for stops, while **L** is employed to mark voicing and nasality.[4] Table 1.3 on the following page

[4]The duality of **L** is intended to explain why nasals do not show phonemic voicing distinctions.

gives a summary of how elements are mapped to phonetic properties, and Tab. 1.4 lists several examples.

	Vowels		Consonants	
Element	Head	Operator	Head	Operator
A	source	target	alveolar	coronality
I	source	target	palatal	palatalization
U	source	target labiality	bilabial labiodental	velarization
H	NA	high tone	NA	fricative aspirated
L	NA	low tone	NA	nasal voiced
ʔ	NA	NA	stop	stop
_	lax	tense	velar	vacuous

Table 1.3: An approximate mapping from GP elements to articulatory properties

r	({_}, **A**)	a	({_}, **A**)
j	({_}, **I**)	i	({_}, **I**)
w	({_}, **U**)	u	({_}, **U**)
g	({ʔ}, _)	ɨ	({_}, _)
s	({**H**}, **A**)	e	({**A**}, **I**)
n	({**L**,ʔ}, **A**)	ɛ	({**I**}, **A**)

Table 1.4: Typologically common phonological expressions for various phonemes

A more recent revision of GP's feature calculus are licensing constraints (Charette and Göksel 1996; Kaye 2000, 2001). They were originally introduced as a further restriction on how elements may combine in order to account for cross-linguistic

Unfortunately, though, there is a well-known counterexample, the phoneme system of Icelandic. In Icelandic, voiced and voiceless nasals are in phonemic opposition, for example in *senda* [ˈsɛntɐ] 'to send' versus *senta* [ˈsɛn̥tɐ] 'to feel'. If we adopt the standard assumption that **L** denotes both voicing and nasality at the same time, voiceless nasals — even though they are a phonetic option due to the loose mapping from PEs to sounds — cannot contrast with their voiced counterparts at a phonological level. If, on the other hand, we maintain that **L** denotes either voicing or nasality but not both at the same time, we need an additional element for voiced nasals, thus rendering them structurally more complex than their voiceless counterparts, which seems counterintuitive. To my knowledge, there have been no attempts to solve this conundrum.

({},**A**)	[ɑ]	({**A̅**},**I**)	[ɛ]	({**Ī**},**A**)	[æ]
({},**I**)	[i]	({**A̅**},**U**)	[ɔ]	({**A̅**,**Ī**},**U**)	[œ]
({},**U**)	[u]			({**Ī**},**U**)	[y]

Table 1.5: The vowel system of Finnish

variation in phoneme inventories. For instance, two licensing constraints suffice to get the full vowel system of Finnish, which is listed in Tab. 1.5 (note that as Finnish isn't a tone language, only **A**, **I** and **U** are licit vowel features).

(1) a. All PEs have a non-empty head.

b. U cannot be an operator.

Practitioners of GP consider the step from over 20 binary to less than 10 privative features with few empirically notable sacrifices in expressivity (although what counts as "notable" is open to interpretation) as one of its most important traits. Thus it is no wonder that — as already indicated at the beginning of this section — there have been numerous modifications in the literature to the feature system described above. Harris and Lindsey (1995), for example, propose a slightly different system comprising **A**, **I**, **U**, **R**, **h**, ʔ and enrich it with both an elaborate feature geometry and acoustic phonetic content (in contrast to the articulatory grounding of SPE's features). Jensen (1994), noting the very restricted utility of ʔ, proposes that the property of being a stop should not be encoded by features but rather by structural means, thereby reducing the set of features to **A**, **I**, **U**, **H** and **L**. A comparable argument is put forth by Pöchtrager (2006) regarding **H**. Finally, Pöchtrager (p.c.) suggests that **A** and **L**, too, are structural in nature. From an SPE perspective, these assertions seem to make little sense, because feature matrices are the only information-encoding structures SPE has at its disposal. So what kind of additional structure are Jensen, Pöchtrager and other Government phonologists talking about?

1.3.2 Phonological Structure

GP takes another hint from autosegmental phonology in assuming that phonology does not operate on mere strings of feature matrices but on highly structured representations that differentiate between *melody*, i.e. elements and PEs, and the phonological *structure*, which the melodic material is attached to. In autosegmental phonology, the structural material consists of a string of nodes called the *skeleton* and the association lines that connect the melodic material to said string. The skeleton effectively establishes a linear order on the phonological representation such that it can be mapped to a sequence of phonetic instructions. GP takes the structural component of autosegmental phonology and combines it with an impoverished version of the familiar syllable template, in which vowels occupy nucleus positions while consonants may appear in onset or coda positions (see Fig. 1.2).

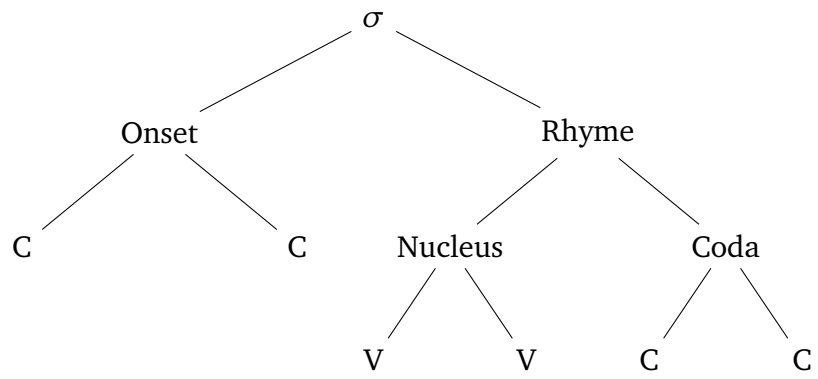


Figure 1.2: A common version of the traditional syllable template

In line with the traditional syllable template, GP requires every skeleton node to be associated to exactly one nucleus, onset or coda. A constituent c with a node x associated to it is said to *dominate* x. GP also leaves the branching factor mostly unaltered for these three constituents: nuclei and onsets may dominate at most two

skeleton nodes, codas, however at most one. For rhymes it is further stipulated that they may branch only if their nucleus does not. Thus no branching nucleus can be followed by a coda, and any such configuration has to be reanalyzed as a coda-less structure. This is formally expressed by the binarity theorem, for which one extends the notion of dominance to the entire syllable template in the obvious way:

(2) *Binarity Theorem*

No constituent may properly dominate more than two skeleton nodes.

Since a rhyme with both a coda and a branching nucleus properly dominates three skeleton nodes, it is ruled out by the Binarity Theorem. Puzzlingly, the theorem also seems to rule out configurations in which σ properly dominates three skeleton nodes: branching nuclei following a (possibly branching) onset, branching onsets preceding a (possibly branching) nucleus, and onset-nucleus-coda configurations. But this problem does not arise in GP, because, speaking in syntactic terms, onsets and rhymes are the roots of distinct trees. Rather than Fig. 1.2, we find Fig. 1.3, where onsets and rhymes aren't connected by a σ-node and codas are analyzed as *rhymal complements*, i.e. as the right daughter of a rhyme.[5]

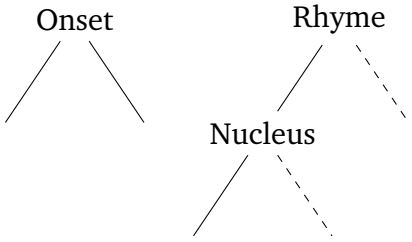


Figure 1.3: GP's modified syllable template (dashed branches may not cooccur)

[5]This minor shift in perspective has the advantage that all remaining constituents — rhymes, onsets and nuclei — have many properties in common, among them their maximum branching factor. Nothing hinges on the rhymal complement analysis of codas, though; for my formalization of GP in Chap. 2, a dedicated coda constituent is in fact the more convenient solution.

But now that onsets and rhymes aren't sisters anymore, additional machinery is required to ensure that not every arbitrary sequence of constituents may qualify as a well-formed phonological structure. This is taken care of by the following constraints.

(3) *Cooccurrence Restrictions on Onsets and Rhymes*

a. Every nucleus may license an immediately preceding onset.

b. Every nucleus is immediately preceded by an onset.

c. Every onset must be licensed by an immediately following nucleus.

d. Every branching rhyme immediately precedes a unary branching onset.

The first three conditions guarantee that phonological expressions consist of sequences of onset-rhyme-pairs, while the latter ensures that words may not end in a coda. This is obviously counter-intuitive considering the abundance of words ending in a single consonant, but I ask the reader to bear with me for now until we reach Subsec. 1.3.4.

The notion of licensing introduced in (3) above can be exploited to enforce another requirement on nuclei.

(4) Every constituent licensor must dominate a skeletal point.

In other words, every nucleus must be associated to a skeletal point. Now this is a truly puzzling constraint. Didn't we already say that constituents are associated to skeletal nodes? Well, no, in fact we only required that for every node there has to be a constituent that dominates it. This doesn't rule out unassociated constituents floating around freely in our phonological representation. Thus it is less the additional restriction on nuclei established by (4) that is noteworthy, than the lack of an analogous principle for onsets. That is to say, onsets do not have to

be associated to skeleton nodes, in stark contrast to nuclei (and codas, which aren't constituents by stipulation).

Summing up the effects of the conditions above, the syllable template reduces to six basic building blocks in Fig. 1.4, which are assembled according to the rules in (5) to yield GP's constituent structure. As an easy exercise, the reader might want to check that the example structures in Fig. 1.5 on the following page are indeed (il)licit.

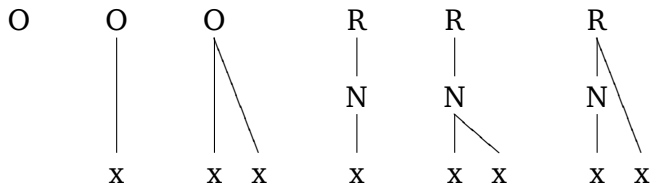


Figure 1.4: The six basic building blocks of phonological structure in GP

(5) *How to combine the building blocks*

a. Every structure consists of at least one rhyme.

b. Every rhyme is immediately preceded by exactly one onset.

c. Every onset immediately precedes exactly one rhyme.

d. Every branching rhyme immediately precedes a unary branching onset.

1.3.3 Melodic Licensing

The licensing conditions of the previous section are located at the level of constituents, or speaking in pictorial terms, above the skeletal nodes. But there are also conditions that apply below them, at the level of *melody*, where the PEs reside. As a rule of thumb, every PE has to be associated to exactly one skeletal node, but skeletal nodes need not be associated to any melodic material. There are several

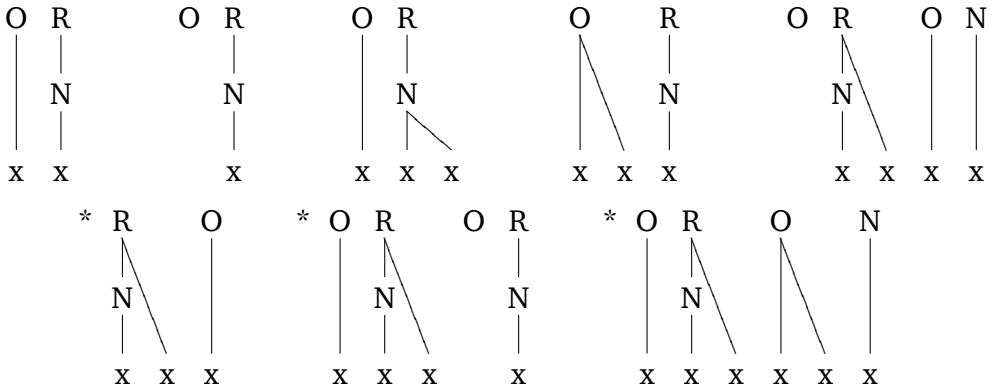


Figure 1.5: Examples of licit and illicit structures

exceptions to this generalization, some of which have to be deferred to the next two sections. For now, we observe only that

- no node may have more than two PEs associated to it, and
- in binary branching constituents, each node must have some PE associated to it, and
- the PE associated to right node of branching onset must be melodically licensed by the PE associated to the left node, and
- the PE associated to right node of a branching rhyme (i.e. the coda) must be melodically licensed by the PE associated to the node of the onset that immediately follows the coda, and
- a node with more than one PE associated to it cannot be melodically licensed.

The first two conditions are straightforward and merely serve in establishing lower and upper bounds on the numbers of PEs per node for specific configurations. The other three invoke the as yet undefined notion of *melodic licensing*. This is merely a technical device that represents the basic fact that certain sequences of

consonants or vowels are licit, while others are not — just think of English *plow* versus **lpow*. Astute readers might object, though, that [lp] is a licit consonant cluster in coda positions, as is witnessed by a plethora of monomorphemic words, foremost *help*. But this is exactly the reason why codas have to be licensed "from the right" instead. To see how this follows, we first need to understand what the GP structures for word-final codas look like, an important point that I left open earlier during the discussion of the syllable template. Figure 1.6 exemplifies how GP reinterprets the canonical syllable template such that what might appear to be a word final coda is, in fact, not word final. It is either an onset followed by a nucleus, or a coda and an onset followed by a nucleus. For as we now allow skeletal nodes to be free of melodic content, this nucleus can remain hidden and hence makes it possible to accommodate all kinds of clusters that previously seemed problematic for GP's insistence on onset-rhyme pairs.

The way GP reinterprets coda clusters makes an empirical prediction: since the licensing of consonant clusters in coda position (i.e. C-O under GP's analysis) is the mirror image of branching onsets, in which it proceeds from left to right rather than the other way round, we predict that an onset cluster is licit in a language if and only if its mirror image is a licit coda cluster. This is borne out for many clusters such as [pl] in English, but there are notable exceptions. In particular, it seems that coda clusters are less restricted than onset clusters. For instance, [mp] is a well-formed coda cluster of English (*camp*, *clamp*, *ramp*, and plenty more), while [pm] is not a licit onset. Apparently, then, the conditions for when a PE is melodically licensed may vary between structural configurations. Unfortunately, very little is known about melodic licensing. So far, a tentative consensus has been reached that for consonants, a PE *q* is licensed by a PE *p* if *p* contains fewer elements than *q*, whereas for vowels, *q* is licensed only if *p* contains the element **A**. I know of no specific proposal that accounts for the differences between onset and

coda clusters, though.

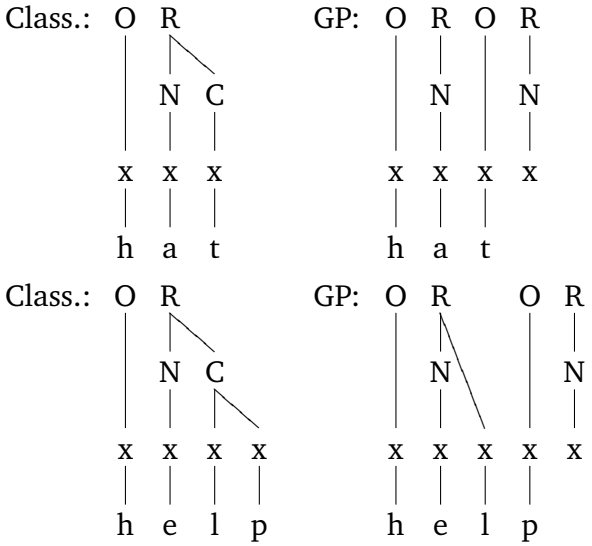


Figure 1.6: GP analysis of consonant clusters and word final consonants

As in the previous section, I deem it prudent to give an explicit list of the configurations that obey the constraints above (Fig. 1.7) so that the reader can simply commit those few structures to his memory and then use them as the basic building blocks for more complex representations.

From the previous example, the reader might have already concluded that branching onsets with exactly one PE associated to each node represent onset clusters. By analogy, then, branching nuclei represent diphthongs. The obvious question, then, is what kind of sounds are represented by two PEs associated to the same node. In the case of nuclei, the answer is well-established: light diphthongs (in return, branching nuclei only represent long diphthongs). For onsets, on the other hand, it has been conjectured that they yield affricates such as in German *Pfropfen* [p͡fʁɔp͡fən] "plug", which is assigned the structure in Fig. 1.8 on page 25. But the mirror image of this sequence is apparently exceedingly rare, with *Karpfen*"carp" being the only example that comes to mind. This one is problematic, however,

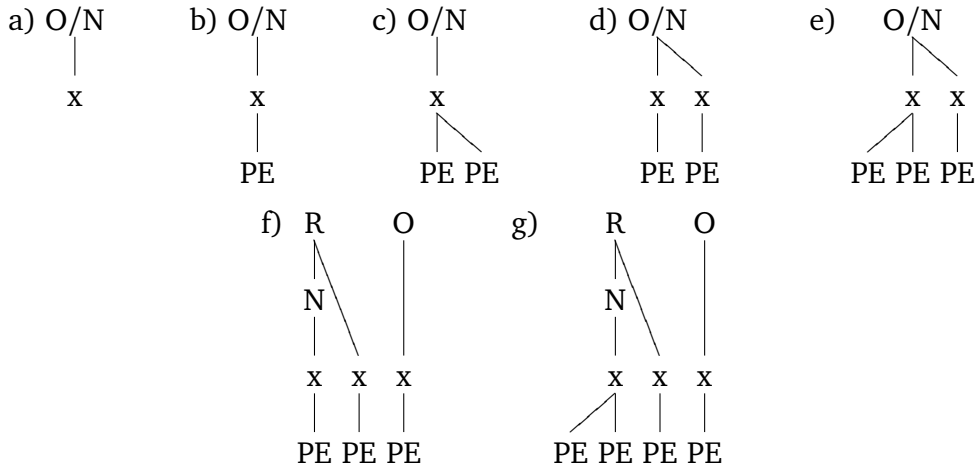
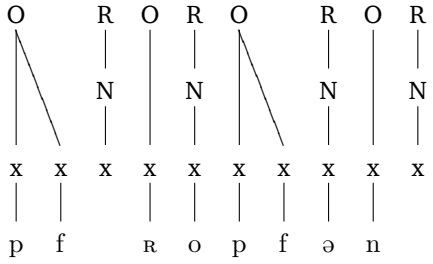


Figure 1.7: Ways of associating PEs to skeleton nodes

because it is usually pronounced [kɑːp͡fən] rather than [kaʀp͡fən]. At least with nasals, though, there are several coda clusters such as *Kampf* "battle"', *Ampfer* "sorrel", *stampfen* "to stomp" and Austrian German *Zumpferl* [t͡sʊmp͡fɐl] "pecker". This might be taken as an indication that the absence of consonant-affricate clusters in coda position is more of a lexical accident than a fact about the structure of affricates. But then again, [p͡fm] is not an attested onset cluster (and sounds very weird to me, significantly stranger than a [ʀp͡f] coda cluster), which suggests that something else than just lexical idiosyncrasies are involved here.[6] In sum, affricates

[6]With the exception of *Kampf*, the structures can moreover be reanalyzed as follows using several empty nuclei.



This analysis does not violate the restrictions on the distribution of empty nuclei that are introduced in the next section.

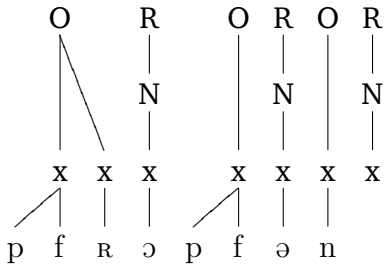


Figure 1.8: A conceivable GP analysis of German [p͡fʁɔp͡fən]

might be analyzed as two PEs attaching to one node dominated by an onset, but it is not an innocent assumption.[7]

With normal vowels and consonsants as well diphthongs and affricates sufficiently covered, there are still two classes to take care of, namely long vowels and geminates. These constitute the exception to the rule of thumb stated at the beginning of this section in so far as they are interpreted as a single PE associated to both nodes of a branching constituent — nuclei for long vowels, onsets for geminates. Examples are given in Fig. 1.9 on the following page, which also showcases some of the other configurations we have encountered.

1.3.4 Empty Categories and p-Licensing

As we just saw, GP allows for certain positions to remain unpronounced, just like traces in syntax. An unpronounced constituent is called an *empty category*, and the use of empty categories is what allows GP to reconcile its restricted syllable template with the wide range of attested syllable types. But obviously the distribution of empty categories has to be carefully regulated lest any kind of syllable structure could be derived in GP as sequences of onsets and possibly empty nuclei. To

[7]Therefore, the formalization in Chap. 2 allows only nodes dominated by a nucleus to be associated to two PEs. However, it can easily be extended that way by removing the reference to onsets in axiom **M2**.

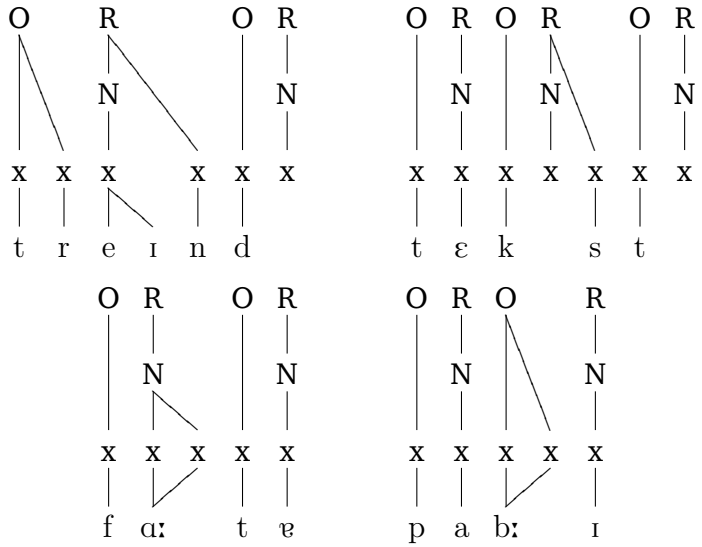


Figure 1.9: Some phonological structures in GP (with IPA notation)

illustrate: no language allows monomorphemic words to contain a cluster of five adjacent consonants, yet if any nucleus may remain unpronounced, we could model such a cluster as a sequence of five pairs consisting of a unary branching onset and an unrealized nucleus. The task of restricting the distribution of empty nuclei is handled by the *Empty Category Principle* (ECP), which is sometimes called the *phonological ECP* to distinguish it from its syntactic counterpart.

(6) *The phonological ECP*

A *p-licensed* empty category receives no phonetic interpretation.

Before we turn to the intricate notion of p-licensing, let me point out that the ECP holds only of empty categories. Whether a nucleus is empty or not, i.e. whether it is associated with a phonological expression is determined in the lexicon but may be altered by (mor)phonological processes later on. In any case, only the nuclei dominating no phonological expressions are subject to the ECP; if a nucleus already dominates a phonological expression, it is immaterial whether it is p-licensed or not, the phonological expression will have to be pronounced in any case. But this

raises the question, of course, what phonetic interpretation could be assigned to an empty category, which by definition has no phonetic content in the first place. The intuitively most appealing conjecture is that languages should pick whatever is their realization of the empty PE, i.e. (_, _). Somewhat surprisingly, this seems to be correct. In Turkish and Arabic, for example, an empty nucleus is realized as [ɨ], whereas German opts for [ə].

Let us return to p-licensing now. An empty category is p-licensed iff it satisfies at least one out of three structural criteria, two of which are fairly simple. The first one is in fact a language-specific parameter, the *Final Empty Nuclei Parameter* (FEN). If the parameter is set to true, any word-final empty nucleus is p-licensed and may thus remain unpronounced. We have already seen several instances of such unpronounced word-final nuclei, e.g. in the analysis of German *Pfropfen* (Fig. 1.8). The second condition is *Magic Licensing*, according to which a nucleus is licensed if it is followed by a coda-onset sequence in which the coda hosts a sibilant. Such a configuration is part of the structure of [tɛkst] depicted in Fig. 1.9 on the previous page. The third condition, *Proper Government*, is significantly more difficult to grasp. Its definition reads as follows:

(7) *Proper Government*

Nucleus *a* properly governs nucleus *b* iff

a. *a* and *b* are adjacent on the relevant projection level, and

b. *a* is not itself p-licensed, and

c. neither *a* nor *b* are government licensors.

As I chose to cut back on terminology in the exposition, not all the technical vocabulary is in place to make sense of these requirements. However, (7) can be rewritten as the noticeably less opaque (8).

(8) *Proper Government* (simplified)

Nucleus *a* properly governs nucleus *b* iff

a. *a* is the first nucleus to the right of *b*, and

b. *a* is not itself p-licensed, and

c. no skeleton nodes between *a* and *b* are involved in any melodic licensing.

Now (8a) and (8b) should be almost self-explanatory, while (8c) simply comes down to the requirement that no branching onset or coda-onset configurations appear between the two nuclei, since these are the configurations where melodic licensing takes place.

The best empirical arguments for Proper Government come from templatic languages like Hebrew, where a lexical item is fully specified by its root, a sequence of 3 or 4 consonants. For example, the root for write is /ktb/. Given GP's method of syllabification, this can be analyzed as three onset-rhyme pairs, where each onset hosts one consonant and the rhymes are lexically empty. Now the interesting catch is that how the root is realized depends only on the presence of further morphological affixes. Take a look at Fig. 1.10 on the following page. In the first row, you see the structures for [ktɨb] "he writes" and [kɨtbu] "they write". The third person singular is not morphologically marked in Hebrew, so [ktɨb] is the pronunciation of the unmodified root. How does this come about? Since proper government proceeds from right to left, we start out at the right edge of the word and move leftwards. The word-final nucleus N_3 is of course the first one we encounter. It contains no lexical material, so in order to see whether it must be pronounced we have to check if it is p-licensed. Assuming that the FEN-parameter is set for Hebrew, it is indeed p-licensed and does not have to be pronounced. Now we move to the left and stop at N_2. The first nucleus to the right of it is N_3, which as we have already verified

is p-licensed. Consequently, N_3 cannot properly govern N_2. Nor is N_2 word final or in a magic licensing configurations. Thus N_2 is not p-licensed. Now if we move further to the left, we encounter the last nucleus, N_1. As N_2, it is neither word-final nor in a magic licensing configuration. However, the first nucleus to its right, that is N_2, is not p-licensed. We also find that the two are separated only by a single unary branching onset, so no melodic licensing takes places between them and N_2 is indeed a proper governor for N_1. Hence the latter remains unpronounced.

In the case of [kɨtbu], the third person plural is represented by the suffix /u/, which moves into the last nucleus of the word. Since said nucleus is now lexically filled, it is not subject to FEN, so it is not p-licensed and properly governs N_2, which thus remains silent. But now that N_2 is p-licensed, it no longer properly governs N_1, and as none of the other conditions for p-licensing are satisfied either, N_1 isn't p-licensed and must be pronounced.

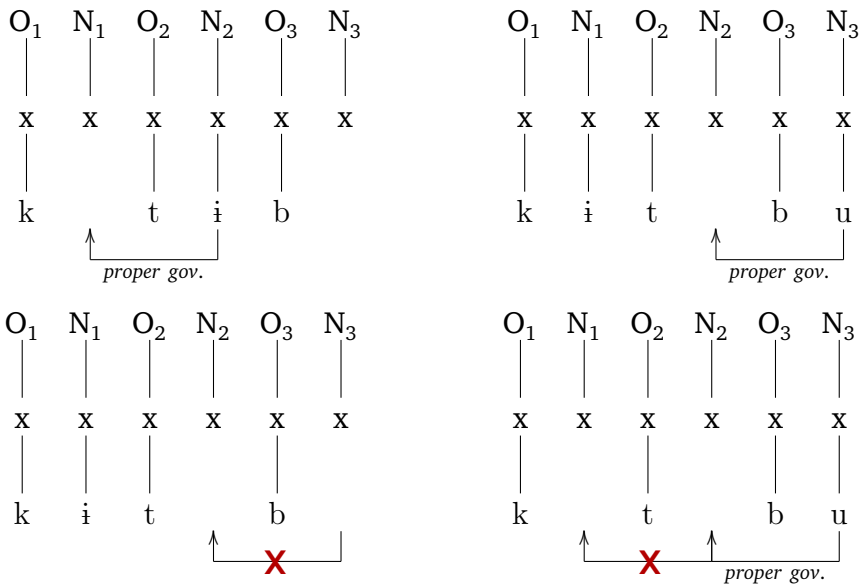


Figure 1.10: Proper government in Hebrew paradigms

An example for the relevance of melodic licesing comes from French (see Charette 1990 for details). The words *ennemi* "enemy", *semaine* "week" and *revenu*

"came back" are pronounced [ɛnmi], [smɛn] and [rəvny], respectively. Given our newly-gained knowledge of proper government, this is hardly surprising. In fact, French seems to behave exactly like Hebrew, as the reader should be able to verify for himself. But for *secret* "secret", the correct pronunciation is [səkrɛ], not [skrɛ]. This is so because [k] occupies a coda position and is melodically licensed by the [r] residing in the onset. Hence the nucleus hosting [ɛ], although it is not p-licensed, cannot properly govern the first nucleus, whence it has to be realized as a schwa.

Putting it all together, we get the definition of p-licensing below.

(9) *p-licensing*

a. Final Empty Nuclei Parameter (FEN)
   Domain-final empty categories are/aren't p-licensed.

b. Magic Licensing
   Sibilant-consonant codas p-license a preceding empty nucleus.

c. Proper Government
   Properly governed (empty) nuclei are p-licensed.

1.3.5 Spreading

So far we have seen a lot of talk about PEs, constituents and various licensing conditions. But the procedural aspects that lay at the core of early generative phonology have been mostly ignored. How does GP account for the kind of interactions between sounds that are so ubiquitous in phonology? The answer is spreading: every element is allowed to associate itself to multiple nodes. A short remark is in order here: Spreading is slightly at odds with what was said in Sect. 1.3.3, namely that entire PEs are associated to skeleton nodes. For the purposes of spreading, it makes sense to reinterpret PEs as a simplified notation for a system with two

Stem	Gloss	Imperative
kɨs	“reduce”	kɨsɨn
kal	“remain”	kalɨn
gir	“enter”	girin
kes	“cut”	kesin
kur	“establish”	kurun
sor	“ask”	sorun
gül	“laugh”	gülün
gör	“see”	görün

Table 1.6: Vowel harmony in Turkish

melodic tiers, one for heads and one for operators, where the individual elements reside and are connected directly to the skeleton node.

Consider the small data set for Turkish vowel harmony in Tab. 1.6(taken from Charette and Göksel 1996). The alternation in the vowel of the suffix can be accounted for if we make three assumptions:

- the nucleus of the suffix is empty, and
- **I** and **U** may spread, and
- **A** may not spread.

The analysis is sketched in Fig. 1.11 on page 33. The general idea is that since the nucleus of the suffix is empty, the elements of the stem can freely spread into it — if they are capable of spreading, that is. In the case of kɨs and kal, there are no elements that could spread into the suffix, and so we get the unaltered form of a pronounced empty nucleus in Turkish, [ɨ].[8] In all the other cases, it suffices to know what elements a sound consist of to predict which vowel will surface in the suffix.

[8]Some readers might wonder why we get [kɨsɨn] rather than [ksɨn]. This might be related to morphology, about the effects of which very little is known in GP (though see Kaye 1995). Kie Zuraw (p.c.) furthermore points out that no words in Turkish are allowed to start with [ks], and word-initial consonant-clusters are fairly rare in general.

Generally, the suffix contains the vowel that is represented by the PE that can be obtained from the PE of the stem vowel by removing **A**, as this is the only feature that does not spread.

Unfortunately, very little is known about the specifics of spreading. As for directionality, spreading can proceed to the left or to the right, but it is unclear if one feature can spread in both directions simultaneously, whether it can change directions after a while, and how far it can spread. It is also not clear from which positions features may spread, and whether certain configurations come with specific restrictions (e.g. if spreading is mandatory from nuclei and optional from onsets). Nor are there claims in the literature concerning the minimum or maximum distance that a feature can (or must) spread. There is also the open question of the status of delinking, i.e. the removal of association lines in order to detach features, which seems to be required for dissimilation and even some cases of assimilation (for instance, as $\mathrm{m} = (\{\overline{\mathbf{L}},\overline{\mathbf{?}}\},\underline{\mathbf{U}})$ and $\mathrm{n} = (\{\overline{\mathbf{L}},\overline{\mathbf{?}}\},\underline{\mathbf{A}})$, assimilation of $/\mathrm{n}/$ to $[\mathrm{m}]$ after $/\mathrm{b}/$ seems to require both spreading of **U** and delinking of **A**). Moreover, many aspects of the interaction of phonology and morphology are not well understood yet in GP. While this is undeniably a shortcoming of GP, it is a defendable one. After all, GP was designed to deal with problems that seemed arbitrary or overly complicated at best from the perspective of SPE or autosegmental phonology. But with a few exceptions like vowel harmony and umlaut, these aren't the kind of problems where spreading has a decisive role to play. The focus of GP has always been on how the interaction of elements, the syllable template and the ECP can explain puzzling paradigms. Without ruining the surprise, I can already tell the reader that the model-theoretic formalization of GP does indeed highlight a difference between these modules on the one side and spreading on the other.

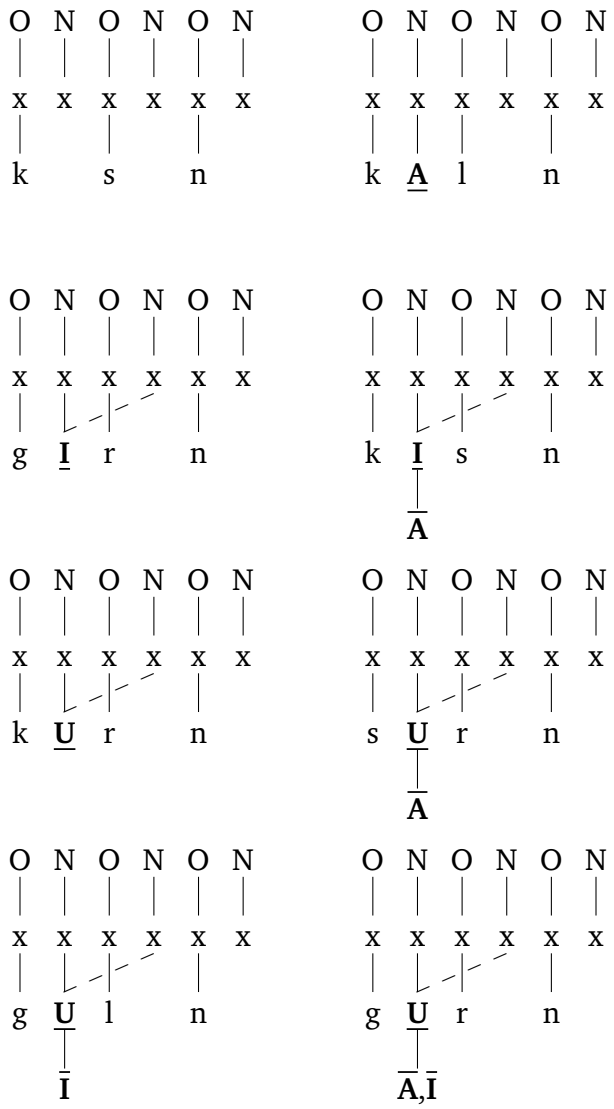


Figure 1.11: Analysis of Turkish vowel harmony

CHAPTER 2

Logical Formalization

2.1 Overview

This chapter is devoted to the logical formalization of GP. Before that, though, I discuss the advantages of an approach grounded in mathematical logic, after which a short introduction to the basics of formal logic is given in Sec. 2.3.

2.2 The Virtues of a Formal Approach

2.2.1 Why Logic?

Linguistic theories aren't monolithic entities; they usually come in numerous flavors that differ to varying degrees from the original proposals. Variants of Optimality Theory (OT), for example, can be built from a vast array of components such as output-output correspondence and sympathy constraints. Clearly, these modifications aren't *ad hoc* inventions but are motivated by empirical concerns, so we should expect them to a have a noticeable impact on the inner workings of the theory. Unfortunately, it is often difficult to see how exactly these changes affect the original theory and the predictions it makes. This creates a big problem for theory comparisons: instead of comparing, say, OT to SPE, different incarnations of OT have to be compared to different incarnations of SPE. A quick survey of the

development of phonology over the last 40 years shows that any well-developed phonological theory has at least three such optional modifications which can be mixed and matched, thereby giving rise to eight variants. Even under optimal conditions, then, a thorough comparison would have to consider at least sixteen theories, truly a herculean task.

An efficient way to reduce the complexity of the comparisons is to group theories into classes from which they inherit certain properties. If these properties are our only concern, it is sufficient to consider only classes instead of all the theories they contain. But what measure should be used as a classification scheme? Ideally, it will be general enough to allow for an easy and reliable classification of specific theories, while at the same time offering enough detail to capture properties of genuine linguistic interest. In this section, I shall try to convince the reader that tools from mathematical logic provide us with a classification mechanism that fulfills both requirements.

Allow me to illustrate the connection between linguistic theories and mathematical logic with an example first. Consider the formula $L \rightarrow \neg H$ of *propositional logic*. It states that the presence of a low tone implies the absence of a high tone. We could just as well phrase the logical formula as a linguistic constraint: "No segment associated with L may be associated with H". Now take a look at the structure in Fig. 2.1 on the following page. It is easy to see that only the leftmost structure obeys the constraint — or as logicians would say, only the leftmost structure *satisfies* $L \rightarrow \neg H$ and is thus a *model* of it. This is also why I call my approach model-theoretic phonology.

We may use additional formulas to impose further well-formedness conditions, just as we can add further rules and constraints to phonological theories. But when we try to write a formula which imposes the requirement that every syllable with a

high tone is both preceded and followed by syllables with a low tone, we run into a problem. Propositional logic cannot do this, as it considers only isolated nodes and fails to take context information into account.[1] This can be fixed by enriching the logic with two *operators* $\triangleleft$ and $\triangleright$, which talk about the nodes immediately to the left and immediately to the right, respectively. Our propositional logic has now become a *modal logic*. The formula $H \rightarrow \triangleleft\triangleleft\, L \wedge \triangleright\triangleright\, L$ then enforces that two steps to the left and two steps to the right of a high tone there is a low tone. By now the reader should be able to check that only the structure in the middle is a model of this formula. Crucially, this implies that none of the structures is a model for both formulas.

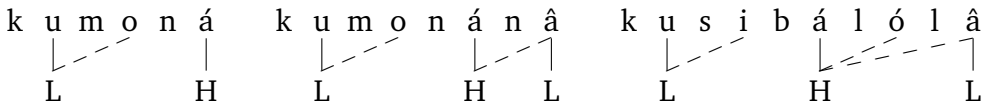


Figure 2.1: The formulas $L \rightarrow \neg H$ and $H \rightarrow \triangleleft\triangleleft\, L \wedge \triangleright\triangleright\, L$ are satisfied only by the leftmost structure and the one in the middle, respectively. None of the three structures satisfies both formulas.

Using more and more formulas as illustrated above, one restricts the set of well-formed structures in the same way a phonological theory does, although certain parts of a theory might require further operators or other modifications to propositional logic. The mathematical literature offers a broad range of logics which can be obtained in this way, the properties of which are well-known. The approach I am advocating here is all about establishing connections between theories and these logics, or putting it slightly differently, about using logics to classify linguistic

[1] If we assume that propositional symbols range not over isolated nodes but rather continuous sequences thereof, propositional logic is indeed capable of talking about structure, too. However, this perspective is not ideally suited to the formalization of most phonological theories, as they posit nodes rather than substrings as the atomic unit whose distribution and surface-realization is to be regulated. In addition, we will see later on that from a modal logic perspective ways of generalizing GP readily suggest themselves, whereas from a purely propositional perspective they arguably wouldn't.

theories. In particular, a theory can be assumed to inherit some of the properties of the weakest logic that is still sufficiently powerful to formalize it.

While the idea seems rather abstract, it is natural and efficient in praxis. Most importantly, the classification of multiple variants requires hardly any additional work. After formalizing the original proposal, e.g. OT with correspondence theory, we are left with the easy task of formalizing the modifications, say, sympathy constraints. If it turns out that the logic used for the original theory is too weak for the altered version, we immediately know that the latter is more powerful than the former. This is exactly what Potts and Pullum (2002) did in their investigation of OT to show that sympathy constraints and output-output correspondence are proper extensions of standard OT. Potts and Pullum's case study also demonstrates that the perspective from mathematical logic affords us new insights that are of immediate linguistic relevance and would be very difficult to obtain with traditional methods based on empirical comparisons. For instance, it allows us to derive *universal insufficiency* results by proving that specific phonological phenomena are beyond the reach of certain classes. That is to say, every phonological theory belonging to class C will fail to account for phenomenon P if P cannot be described in the logic corresponding to C. The traditional approach, on the other hand, derives *theory-specific sufficiency* results by devising an account of a specific phonological phenomenon in the theory under scrutiny. Rather than a replacement, then, the logical approach is a useful complement to the traditional one.

That it is no substitute for thorough empirical comparisons is also witnessed by the fact that the gains from a deliberate restriction to classes of theories come at the cost of reduced granularity. Hence MTP fairs better than alternative approaches in getting the big picture right; it derives general, broadly applicable results pertaining to generative capacity, computational complexity, memory requirements and parsing.

It has less to say about technical minutiae that do not correspond to class distinctions. This does not mean that we cannot use MTP for the investigation of such details, but it would be just as laborious a task as with any other approach.

It should also be pointed out that the way model-theoretic syntax employs logic differs significantly from earlier logical approaches, such as Stabler (1992). These approaches are proof-theoretic in nature, which means that they construe a linguistic theory as a set of axioms and the language licensed by said theory as the set of logical theorems that can be proved starting from these axioms. As a metaphor (a clumsy one, as metaphors usually are), linguists may think of this as a derivational perspective, where we start out with some primitives, e.g. Merge, Move and the lexical items, and then try to construct the desired tree. Such an approach is particularly helpful in determining how different parts of a theory interact. For instance, Stabler (1992) showed that the Barriers framework of Chomsky (1986) still derives CED effects as intended if one removes Subjacency from the theory, but not if the ECP is dispensed with; this refutes claims to the contrary made by Chomsky, which were also challenged by Browning (1989) on empirical grounds. Clearly, then, this approach has its merits, too. However, for my project, a model-theoretic approach is easier to handle.[2] It directly represents the structures licensed by a theory, and thus makes the connection between linguistic structures and linguistic theories more tangible. In particular, we may explicitly restrict our attention to the class of mathematical objects that we deem relevant to the enterprise (in our case, strings rather than trees or even more complex graphs) and not worry about whether the logic is capable of picking out this mathematical class all by itself. This is beneficial because we are interested in

[2]There is a philosophical issue here as to whether a scientific theory should be thought of as a set of basic statements or a class of models. While this is an interesting point to ponder on an epistemic and methodological level, it is of no relevance to the results reported herein and will be ignored.

linguistic rather than purely mathematical aspects that make a linguistic theory more complex than another one — that string structures cannot be axiomatized in various logics is unlikely to prove insightful to linguists. By keeping an eye on the licensed structures, it is also easier to figure out what pieces of a theory are but notation without relying on advanced computational tools such as automatic theorem provers. This is important when considering non-standard logics for which such tools are not readily available. Many of the logics we will encounter in later sections are such non-standard logics.

2.2.2 Why not Automata Theory?

Readers that show at least a cursory familiarity with computational phonology might wonder why I choose to advocate logic to the detriment of better established formal tools such as automata theory. First of all, it should be noted that I am not arguing against automata theory as such. Logic has rich connections to automata theory (and, by extension, formal language theory), and I do make use of them several times in this thesis, foremost in Chap. 3. Thus I do not argue against automata theory but rather against the use of automata for the formalization of theories. There are several reasons for that, all of which can be traced back to differences as to what a formal approach to phonology is thought to accomplish. For most computational phonologists, formalization is all about laying the foundations for computationally tractable software implementations, where the gold standard of "computational tractability" is computability by finite-state devices. But as will become evident later on, finite-state devices — restricted as they might be in terms of expressivity — are much more powerful than what is needed for most areas of phonology. For our purposes, this means that virtually all phonological theories belong to the same class when considered from the perspective of finite-state

automata.

Admittedly there are automata-theoretic characterizations for weaker classes, too, but in my opinion these are somewhat cumbersome to work with; more importantly, the multiplication of different automata models only serves in aggravating a general problem of automata-based formalizations, the lack of succinctness and modularity in comparison to the model-theoretic approach.[3] This is best illustrated by a simple example. In Fig. 2.2 on page 42, the automata for two simple modal formulas are given. The formulas are $N \rightarrow \triangleleft O \vee \triangleleft N$ — every nucleus is preceded by an onset or a nucleus — and $O \rightarrow \neg \triangleleft O \vee \neg \triangleright O$ — no onset is both preceded and followed by an onset (i.e. onsets are at most binary branching). A string is accepted by the automaton if one can start at the initial state (the one marked with an incoming arrow) and then, moving from left to right through the string, follow the arrows labeled with the corresponding symbols in the string and end in a final state (indicated by a double circle). So a string like ONNO is accepted by the first automaton, because starting in state 1 we can first move along an O-labeled branch that takes us to state 2, wherefrom we proceed along the N-labeled branch into state 3. From there the second N leads us back to state 1, and there we can once again follow an O-branch, but this one takes us from 1 back to 1 itself. Now we are done scanning the string and the last state we reached is a final state, so the string is accepted by the automaton. Note that we could have taken different routes, some of which might not have ended in a final state, but this is immaterial. As long as

[3] Mathematical succinctness results are well-known. Consider monadic second-order logic (MSO) over strings. We know that every MSO formula can be translated into a finite-state automaton. We also know that this formula may be non-elementarily more succinct than the corresponding automaton — each quantifier may induce an exponential blow-up in the number of states. Crucially, the translation procedure is optimal, that is to say, there is no smaller automaton that would be equivalent to the formula (Meyer 1975). As for the modularity, this problem could possibly be circumvented by decomposing the automata in the spirit of Krohn and Rhodes (1965), but this is a very difficult task and the result would presumalby still be significantly more complicated than the logical formulas we will encounter here.

there is at least one route leading to an accepting state, the string is accepted. A string like NN, on the other hand, is always rejected, because there is no N-labeled branch that leaves the initial state.

As the reader might have noted working through this simple example, the automata, albeit rather simple, fail to make fully explicit the restrictions the constraints imposed on the language, because they give a complete encoding of all the strings that obey the constraint. For instance we find references to codas even though those are never directly mentioned in the constraints. Their presence only follows from extra assumptions about GP constituents. Note that this means that a purely automata-theoretic approach struggles to express constraints in a theory-neutral fashion, it also embodies other assumptions of the theory, whence it is more difficult to port the formalized constraint from one proposal to another. And while these shortcomings might appear manageable for these small automata, just taking their intersection (i.e. enforcing both constraints) yields an automaton that is supremely opaque (see once more Fig. 2.2). Whoever is capable of taking a glance at said automaton and effortlessly extracting the well-formedness conditions it encodes is worthy of the highest appraisal. In comparison, it only takes a passing familiarity with modal logic to get the gist of the corresponding formulas.

In summing up, logic provides us with a tool that allows us to investigate classes of theories rather than specific incarnations thereof, and it does so in an intuitive, easily accessible way that at the same time allows us to relate phonology to areas such as formal language theory and complexity theory, which are of utmost importance in determining the expressivity and parsing properties of those theories. Alternative approaches that are deeply entrenched in computational phonology, foremost automata theory, can be employed in the same way, but they are more cumbersome to use and do not scale well with the number of constraints as they

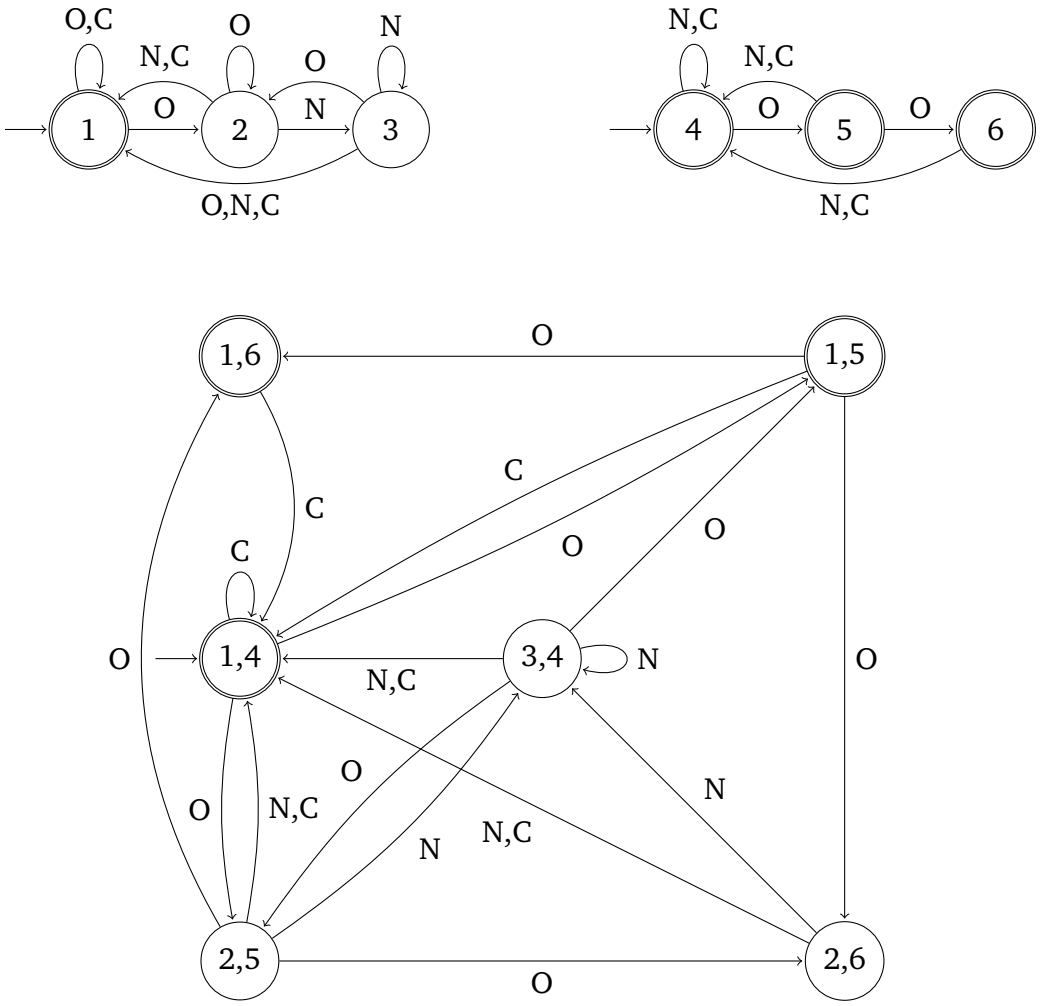


Figure 2.2: Finite-state automata for $N \rightarrow \lhd O \vee \lhd N$ and $O \rightarrow \neg \lhd O \vee \neg \rhd O$), and their intersection (with inaccessible states removed)

are “overly explicit”, i.e. their mode of representation encodes more information than is necessary for our purposes.

2.3 Logic — A Mathematical Primer

Before venturing any further into the mathematical details of the formalization of GP, it seems advisable to look a little closer at what a logic actually is. The previous section already established several intuitive notions and concepts, foremost that there are in fact different kinds of logics and how a logical formula may represent linguistic constraints. These issues are related to what is commonly referred to as the *syntax* and the *semantics* of a logic. The syntax specifies what sequences of formal symbols belong to the logic. From a syntactic perspective, then, a logic is simply a formal language over some alphabet. Let us start with the syntax of propositional logic. Assume we are given some set P of *proposition letters* (also called *propositional variables*). Proposition letters may stand for features, constituents, sounds, strings, or maybe something completely different; for now we don’t commit ourselves to any particular interpretation. Then the propositional logic $\mathscr{L}_P$ is the set of strings that can be built up inductively as follows:

- If p is a proposition letter in P, then p belongs to $\mathscr{L}_P$.

- If ϕ and ψ are strings of $\mathscr{L}_P$, then the following strings also belong to $\mathscr{L}_P$.

$(\neg\phi)$	not ϕ
$(\phi \wedge \psi)$	both ϕ and ψ
$(\phi \vee \psi)$	ϕ or ψ or both
$(\phi \rightarrow \psi)$	ϕ implies ψ
$(\phi \leftrightarrow \psi)$	ϕ and ψ imply each other

So the strings p, $(\neg p)$ and $((\neg(p \wedge q)) \rightarrow (\neg q))$ would belong to $\mathscr{L}_P$ (provided p and q are proposition letters of P), whereas $\neg$ or $(p\neg \wedge q)$ would not.

Logicians usually refer to the strings in $\mathscr{L}_P$ as *(propositional) formulas* or simply *proposition*, and we shall follow this convention. Propositional logics represent the barest kind of logic, and they can be extended in various ways. The best-known extension is first-order logic, in which variables and operators quantifying over those variables are introduced. The technical machinery of first-order logic can be rather daunting for the uninitiated, but fortunately we will operate with an extension of propositional logic that is much weaker and thus easier to master, namely modal logic. Nonetheless familiarity with some foundational facts of first-order logic is required to understand some of the more technical remarks in Chap. 3, whence I briefly present them here.

First-order logic extends propositional logic by adding variables, predicates over those variables, the quantifiers $\exists$ "there is at least one" and $\forall$ "for every" over those variables, and square brackets that indicate the scope of the quantifiers. A formula like $\forall x[P(x) \rightarrow \exists y[\neg Q(y)]]$ can be read as "for all x it holds that if P is true of x, then Q does not hold of some y". Most readers certainly will have encountered such formulas before in linguistics. The new tidbit of mathematical background

knowledge that has to be committed to memory is that one can distinguish fragments of first-order logic depending on the number of variables that they allow to appear in one formula. For instance, the two-variable fragment of first-order logic, abbreviated FO^2, allows only formulas with two variables. So the formula above is an FO^2 formula, whilst a formula like $\exists x \exists y \exists z [P(x) \wedge Q(y) \wedge \neg P(z) \wedge \neg Q(z) \wedge x \neq y \wedge x \neq z \wedge y \neq z]$ belongs to FO^3 (this formula states that there are distinct x, y, and z such that P is true of x, Q is true of y, and neither is true of z).

Whereas first-order logic looks significantly different from propositional logic, all one has to do to turn a propositional logic into a modal logic is add a number of operators. Let O be the set of these operators. Then we add the following clause to the syntactic definition above.

- If $\langle o \rangle$ is an operator belonging to O and ϕ is a string of $\mathscr{L}_p$, then $(\langle o \rangle \, \phi)$ is a string of $\mathscr{L}$.

Let's work through a short example. Suppose we are given the operators $\lhd$ and $\rhd$, and further the proposition letters p and q. Then the strings $(\lhd p)$, $(\neg(\lhd (p \wedge q)))$ and $((\lhd p) \wedge (\rhd q))$ all belong to $\mathscr{L}$, but $(\lhd)$ and $(\lhd p)(\rhd q)$ do not. Strictly speaking, $p \wedge q$ isn't a well-formed string either, since it lacks the outermost brackets, but it is common usage to drop redundant brackets wherever possible, such that the formulas above may be rewritten as $\lhd p$ or $\lhd p \wedge \rhd q$. However, $(\neg(\lhd (p \wedge q)))$ may not be rewritten as $\neg \lhd p \wedge q$, as this would correspond to the formula $(\neg \lhd p) \wedge q$. Thus one has to be careful to drop brackets only according to the following scope rules:

- Negation and modal operators are stronger binders than any other connective.
- The connectives $\wedge$ and $\vee$ are stronger binders than $\rightarrow$ and $\leftrightarrow$.

Thus a formula like $\neg p \vee (\lhd \neg p \wedge q \rightarrow \rhd (q \vee p))$ corresponds to the fully bracketed formula $((\neg p) \vee (((\lhd (\neg p)) \wedge q) \rightarrow (\rhd (q \vee p))))$. There are also rules that regulate how formulas like $p \wedge q \vee \rhd q$ or $p \rightarrow q \leftrightarrow p$ is to be disambiguated, but for the sake of readability I will always make the relevant scope relations explicit through the use of brackets.

Now that we know how to write formulas of a modal logic, we have to specify what they mean, i.e. the semantics of the logic. The general idea is that a formula is evaluated with respect to some structure by first determining which propositions are true at which nodes in the structure (e.g. an SPE feature like -consonantal is true at every segment of the string that isn't a consonant) and then decomposing the connectives and modal operators into a sequence of instructions as to which nodes have to satisfy which propositions. For instance, $p \rightarrow \lhd q$ will be true iff for every node n at which p holds, q is true at some node that is related to n by $\lhd$. If the interpretation of $\lhd$ is "take one step to the left", this formula states that every node satisfying p has a node to its left that satisfies q.

The formal setup proceeds as follows. First we are given a *frame*, which represents the bare bone structure without any information about which propositions are satisfied where. For syntax, say, the frames would be unlabeled trees. If our syntactic theory furthermore makes the assumption that no node may have more than two daughters, the frames would be trees that are at most binary branching. In this paper, our frames will be strings, i.e. a set of nodes that is linearly ordered by the right successor relation. In fact, we will even assume that the frames are *bidirectional*, which means that we can also follow the order defined by the relation in the other direction, such that we effectively have both a right successor and a left successor relation. The relations are then associated to the operators of our modal logic. For example, $\rhd$ would be associated to the right successor relation such that

it can be read as the instruction "move to the right successor of whatever node you are currently at". Frames are then turned into *models* by adding a *valuation*, a function that tells us for each proposition letter at which nodes in the frame it is satisfied. It is this process of mapping proposition letters to nodes that establishes their meaning. Crucially, several proposition letters may be true at the same node, so they are more like features rather than labels.

In mathematical terms, this reads as follows: A frame is an $(n+1)$-tuple $\mathfrak{F} := \langle D, R_i \rangle_{1 \leq i \leq n}$ (read $\mathfrak{F} := \langle D, R_1, R_2, \ldots, R_{n-1}, R_n \rangle$), such that D is a non-empty set of nodes and each $R_i \subseteq D \times D$ is a binary relation between nodes of D. A model is a pair $\mathfrak{M} := \langle \mathfrak{F}, V \rangle$, where the valuation V maps propositional variables to subsets of D. Given sets $O := \{\langle o_i \rangle\}_{1 \leq i \leq n}$ of modal operators and P of propositional variables, a formula ϕ is satisfied by $\mathfrak{M}$ iff it is true at every node w of the domain, where truth is determined according to the rules below:

$\mathfrak{M}, w \models p$	iff	$w \in V(p)$
$\mathfrak{M}, w \models \neg\phi$	iff	$\mathfrak{M}, w \models \phi$ is false
$\mathfrak{M}, w \models \phi \wedge \psi$	iff	both $\mathfrak{M}, w \models \phi$ and $\mathfrak{M}, w \models \psi$ are true
$\mathfrak{M}, w \models \phi \vee \psi$	iff	$\mathfrak{M}, w \models \phi$ or $\mathfrak{M}, w \models \psi$ is true (or both)
$\mathfrak{M}, w \models \phi \rightarrow \psi$	iff	if $\mathfrak{M}, w \models \phi$ is true, then $\mathfrak{M}, w \models \psi$ is also true
$\mathfrak{M}, w \models \phi \leftrightarrow \psi$	iff	$\mathfrak{M}, w \models \phi$ is true iff $\mathfrak{M}, w \models \psi$ is true
$\mathfrak{M}, w \models \langle o_i \rangle \phi$	iff	there is a node v related to w by R_i such that $\mathfrak{M}, v \models \phi$

Some authors (including myself) prefer to save some lines by using a neat trick to reduce the number of connectives. One can show that the four logical connectives of propositional logic can be reduced to the already familiar implication connective $\rightarrow$ and the so-called *falsum* $\bot$, a special proposition that is false at every node in every structure. The formula $\neg\phi$, for instance, can be rewritten as $\phi \rightarrow \bot$. This way one can use all the four connectives but only give the semantics for $\rightarrow$ and $\bot$.

The latter is simply defined by stipulating that $\mathfrak{M}, w \models \bot$ is always false.

This closes our discussion of the mathematics underlying the formalization we are about to begin in the next section. At this point the reader should return to Fig. 2.1 on page 36 and write out a step by step computation that shows that the formula $H \rightarrow \lhd\lhd L \wedge \rhd\rhd L$ is satisfied only by the structure in the middle. It may safely be assumed that the models are defined over bidirectional frames, but one has to be explicit about the set of proposition letters P as well as the valuation V. As soon as those are suitably defined, the rest is a fairly simple mechanical procedure; just keep in mind that a formula is satisfied by a model iff it is true at every node of the model.

2.4 Formalization

2.4.1 Reinterpreting GP-Structures as Strings

Before diving headfirst into a pile of formulas, it is a good idea to get an intuitive idea of what our models will look like. The first step of the formalization is to accommodate GP's feature system. Recall (or go back to Tab. 1.4 on page 15 to refresh your memory) that GP replaces SPE's feature matrices by a pair consisting of a set of privative features, called operators, and a single privative feature, which functions as the head. In my formalization, head and operator features are distinct propositions that are generated from a unique set of base features. Given a GP theory with three features **A**, **I**, **U**, one would use three "head features" $\underline{\mathbf{A}}$, $\underline{\mathbf{I}}$, $\underline{\mathbf{U}}$ and three "operator features" $\overline{\mathbf{A}}$, $\overline{\mathbf{I}}$, $\overline{\mathbf{U}}$. This makes it possible to regulate the entire feature calculus using only propositional logic. Given the six features just listed, the pair for the sound $\varepsilon = (\{\mathbf{I}\}, \underline{\mathbf{A}})$, for instance, is represented by the logical formula $\underline{\mathbf{A}} \wedge \neg\underline{\mathbf{I}} \wedge \neg\underline{\mathbf{U}} \wedge \neg\overline{\mathbf{A}} \wedge \overline{\mathbf{I}} \wedge \neg\overline{\mathbf{U}}$. It is also straightforward to enforce the uniqueness of

the head feature by formulas such as $\underline{\mathbf{A}} \rightarrow \neg \underline{\mathbf{I}} \wedge \neg \underline{\mathbf{U}}$ for every head feature. Similarly, formulas such as $\underline{\mathbf{A}} \rightarrow \neg \overline{\mathbf{A}}$ ensure that a feature, in this case **A**, does not occupy both a head and an operator position.

Next we turn the syllable template, which is built around four types of constituents: onsets (O), rhymes (R), nuclei (N) and codas (C), where N and C are dominated by the rhyme. The constituents have to appear in a certain linear order, and the structural configuration they appear in determines whether they are allowed to branch. The precise rules were discussed in Sec. 1.3.2. As in the example in Fig. 2.1 from Sec. 2.2.1, propositional logic is too weak to express such structural information. But we can use a modal logic with operators $\triangleleft$ and $\triangleright$ to take the neighborhood of a node into account. With dedicated propositions for N, O and C (and optionally also R), this logic is capable of expressing all relevant structural constraints. The condition that every coda is followed by an onset, for example, can be rendered as the formula $C \rightarrow \triangleright O$.

There are two special cases that need to be taken care of, though. The first one concerns onsets that are not associated to any skeleton node. This rather rare configuration is used to explain certain phenomena pertaining to word initial /h/ in French (the distinction between *h muet* and *h aspiré*, to be precise). For mathematical reasons, I model this as a normal unary branching O — i.e. an O associated to a single skeleton node — which in turn hosts a special feature `fake`. The feature `fake` tells us that the onset in question represents an unassociated onset. Therefore, restricting the distribution of unassociated onsets is tantatmount to restricting the distribution of the feature `fake`, a simple task. The second minor complication is due to binary branching constituents, which I encode as two adjacent unary branching constituents of the same type. This does not introduce any conceptual confusions since such configurations cannot normally arise in GP,

whence it is safe to assume that they represent binary branching constituents. Note that I am driven to this move by considerations of mathematical simplicity and elegance, but none of my results hinge on these minor alterations.

With these modifications, we get bare GP syllable structures that look like the ones in Fig. 2.3.[4]

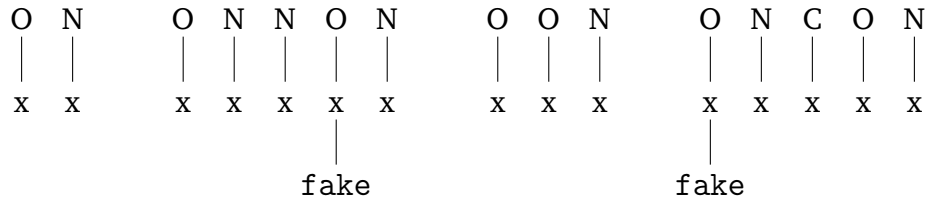


Figure 2.3: Examples of syllable structures in simplified notation

It might be at this point that government phonologists start to take issue with my formalization and the slight simplifications it embodies; the subsequent treatment of empty categories will in all likelihood raise even greater concerns. The definitions and constraints involved in the distribution of empty categories are rather complex, but if we ignore for a moment domain-final nuclei and magic licensing configurations, the underlying intuition is easy to express in the simplified template: If a nucleus is pronounced, the preceding nucleus may remain unpronounced. If a nucleus is not pronounced, the preceding nucleus has to be pronounced. However, if the two nuclei are separated by two or more skeleton nodes associated to O or C, both have to be pronounced under all circumstances. Disbelieving readers may want to check for themselves that this formulation yields the same results as the original definition of the Proper Government condition from Sec. 1.3.4. The reason

[4]GP practitioners might wonder how I accommodate short diphthongs, which are represented by two distinct phonological expressions, say, one for [ɑ] and one for [ɛ], associated to the same skeleton node. I choose to handle this within the feature calculus by introducing another feature parameter (the first one being the distinction between heads and operators) that tells us whether a feature belongs to the first or the second expression.

is that the nuclei are farther apart only if a branching onset or an onset licensing a coda intervenes, and these are exactly the cases where proper government is interrupted. As soon as this is known, we only have to introduce two diacritic features ✓ and μ (read *mute*) that tell us, respectively, if a nucleus is p-licensed and if it is pronounced.

Admittedly, though, a single result that shows how these conditions can be simplified given a different encoding of the syllable template is insufficient to dispel doubts about the faithfulness of the formalization. Even though it is foremost an issue of philosophy of science to determine which parts of a theory need to be represented explicitly in its technical machinery, I believe my modeling decisions can be sufficiently supported on purely pragmatic grounds alone. For one has to keep in mind that the goal is to use as weak a logic as possible; but the weakest logic that might be expressive enough to allow for a direct translation of the conditions involved in proper government, the *two variable fragment of first-order logic* (FO^2), is significantly more powerful than the modal logic I propose to use. One could of course try to put further restrictions on FO^2 to push it down to the level of a modal logic, but there is nothing to be gained from such a cumbersome move, because the two logics would then be identical for our purposes. In sum, the underlying issue here is that a formal approach always has to reconcile linguistic faithfulness with mathematical desiderata if it wants to be useful; the changes to GP I adopt above are, in my opinion, the best compromise between those two poles. Even if some of the readers may still take issue with the slight deviance of our formalization, they can rest assured that it is immaterial for the claims made in this paper, thanks to the granularity of the properties we are interested in.[5]

The last module to be formalized is spreading. Here I employ once more the

[5]Moreover, GP actually benefits from these slight re-encodings, as we would otherwise be pressed to postulate that a very expressive logic is necessary for modeling GP when, in fact, it is not.

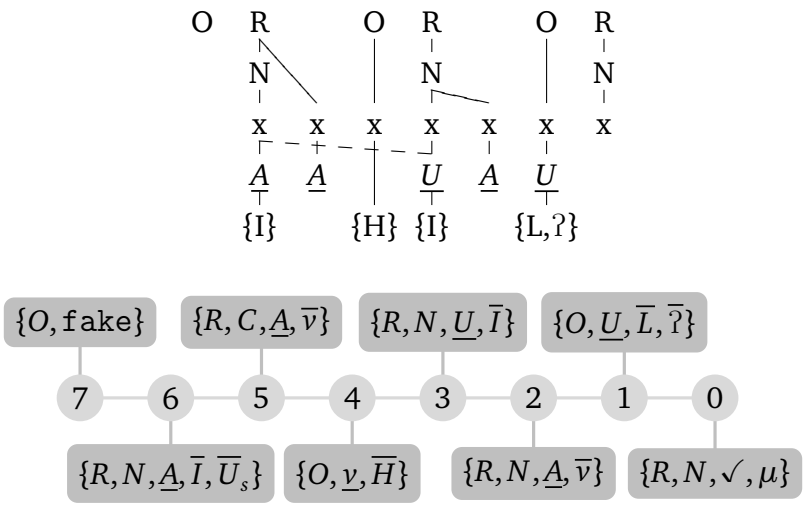


Figure 2.4: Comparison of GP structure and formal model

strategy of adding subscripts to features, this time in order to distinguish *spread* from *local* features. Marking the former with a subscripted s (and indicating empty heads or operators by the symbol v), our final models will look like the example in Fig. 2.4.

2.4.2 Logical Formalization

The material in this section is significantly more technical than anything else we have encountered so far, and the reader may find himself at a loss as to what is crucial information and what but minor technicalities. Hence I have decided to highlight only those points that flesh out or extend the intuitions that I tried to convey in the previous sections. It follows that if some definition or constraint is not elaborated upon, I do not deem it indispensable for a full understanding of the formalization and it will be of interest only to readers that find pleasure in the math itself.

As was established several times by now, I use a modal logic that uses two

modal operators corresponding to "one step to the left" and "one step to the right". Logicians may think of it as the result of removing the "sometime in the future" and "sometime in the past" modalities from restricted temporal logic (Cohen et al. 1993; Etessami et al. 1997). To this audience it might also be of interest to know that the tree model property of modal logic implies that my logic is too weak to define the intended class of models, so we are in fact dealing with a formal description rather than a proper axiomatization. That is to say, the axioms given below are compatible with many models, most of which are not the kind of structure envisioned by GP. Only if we restrict our attention to string models do the axioms constrain the models as desired.

The first step is to multiply out the feature system as indicated in the previous section. Let E be some non-empty finite set of *basic elements* different from the neutral element v (which represents the empty set of GP's feature calculus). We define the set of *elements* $\mathscr{E} := (E \times \{1,2\} \times \{head, operator\} \times \{local, spread\}) \cup (\{v\} \times \{1,2\} \times \{head, operator\} \times \{local\})$. The intended role of the *head*/*operator* and *local*/*spread* parameter is to distinguish elements according to their position in the phonological expression (PE) and whether they arose from a spreading operation, respectively. The second projection is of very limited use and required only by GP's rendition of light diphthongs as two PEs associated to one node in the structure (see fn. 4 on page 50).

Now we add a small number of diacritic features to the mix. The set of melodic features $\mathscr{M} := \mathscr{E} \cup \{\mu, \texttt{fake}, \checkmark\}$ will be our set of propositional variables. The intent is for μ (remember that this is mnemonic for *mute*) and $\checkmark$ to mark unpronounced and licensed segments, respectively, while `fake` denotes an unassociated onset. For the sake of increased readability, the set of propositional variables is "sorted" such that $x \in \mathscr{M}$ is represented by m, $m \in \mathscr{E}$ by e, heads by h, and operators by o.

The variable e_n is taken to stand for any element such that $\pi_2(e) = n$, where $\pi_i(x)$ returns the i^{th} projection of x (for instance $\pi_2(\langle a, b, c\rangle) = b$). On rare occasions, I write $\underline{e}$ and $\overline{e}$ for a specific element e in head and operator position, respectively.

Furthermore, there are three nullary modalities[6], N, O, C, the set of which is designated by $\mathscr{S}$, read *skeleton*. In addition, we introduce two unary diamond operators $\triangleleft$ and $\triangleright$. The set of well-formed formulas is built up in the usual way from $\mathscr{M}$, $\mathscr{S}$, $\triangleleft$, $\triangleright$, $\rightarrow$ and $\bot$ (see Sec. 2.3).

Our intended models $\mathfrak{M} := \langle \mathfrak{F}, V\rangle$ are built over bidirectional frames $\mathfrak{F} := \langle D, R_i, R_\triangleleft\rangle_{i\in\mathscr{S}}$, where D is an initial subset of $\mathbb{N}$, $R_i \subseteq D$ for each $i \in \mathscr{S}$, and $R_\triangleleft$ is the successor function over $\mathbb{N}$. The valuation function $V : \mathscr{M} \rightarrow \wp(D)$ maps propositional variables to subsets of D. The definition of satisfaction is standard, though it should be noted that our models are "numbered from right to left". That is to say, $0 \in D$ marks the right edge of a structure and $n+1$ is to the left of n. This is the easiest route due to GP's proper government being computed from right to left.

$$
\begin{array}{lll}
\mathfrak{M}, w \models \bot & & \text{never} \\
\mathfrak{M}, w \models p & \text{iff} & w \in V(p) \\
\mathfrak{M}, w \models \phi \rightarrow \psi & \text{iff} & \mathfrak{M}, w \models \phi \text{ implies } \mathfrak{M}, w \models \psi \\
\mathfrak{M}, w \models N & \text{iff} & w \in R_N \\
\mathfrak{M}, w \models O & \text{iff} & w \in R_O \\
\mathfrak{M}, w \models C & \text{iff} & w \in R_C \\
\mathfrak{M}, w \models \triangleleft \phi & \text{iff} & \mathfrak{M}, w+1 \models \phi \\
\mathfrak{M}, w \models \triangleright \phi & \text{iff} & \mathfrak{M}, w-1 \models \phi
\end{array}
$$

[6] I follow the terminology of Blackburn et al. (2002) here. Nullary modalities correspond to unary relations and can hence be thought of as propositional constants. As far as I can see, nothing hinges on whether we treat constituent labels as nullary modalities, propositional constants, or propositional variables; my motivation in separating them from phonological features stems solely from the parallel distinction between melody and constituency in GP.

With the logic fully defined, we can turn to the axioms for GP. The formalization of the skeleton is straightforward with the assumptions introduced in the previous section, i.e. that one models binary branching constituents as two adjacent unary branching ones and views rhymes as mere notational devices. Recall that Ns containing light diphthongs are implemented as a single N with both e_1 and e_2 elements associated to it.

S1	$\bigwedge_{i\in\mathscr{S}}(i \leftrightarrow \bigwedge_{i\neq j\in\mathscr{S}} \neg j)$	Unique constituency
S2	$(\neg \lhd \neg\bot \rightarrow O) \wedge (\neg \rhd \neg\bot \rightarrow N)$	Word edges
S3	$R \leftrightarrow (N \vee C)$	Definition of rhyme
S4	$N \rightarrow \lhd O \vee \lhd N$	Nucleus placement
S5	$O \rightarrow \neg \lhd O \vee \neg \rhd O$	Binary branching onsets
S6	$R \rightarrow \neg \lhd R \vee \neg \rhd R$	Binary branching rhymes
S7	$C \rightarrow \lhd N \wedge \rhd O$	Coda placement

Axioms **S1** and **S2** are slightly more complicated than **S3**–**S7**, for different reasons. In **S1**, a notational trick is made use of in the form of big connectives in order to shorten the formula. Big connectives are similar to quantifiers in that they bind a variable and instantiate it with different values. In the case at hand, there are two variables, i and j, which are instantiated such that i is some element of $\mathscr{S}$ (i.e. N, O or C) and j some element of $\mathscr{S}$ distinct from i. One then has to go through all possible instantiations of the variables and connect the generated formulas by the small counterpart of the big connective. For example, if $i = N$, we get the formula $\phi := N \leftrightarrow \neg O \wedge \neg C$, for $i = O$ the formula $\psi := O \leftrightarrow \neg N \wedge \neg C$, and for $i = C$ the formula $\rho := C \leftrightarrow \neg N \wedge \neg O$. Thus the fully expanded version of **S1** corresponds to $\phi \wedge \psi \wedge \rho$.

The difficulty in **S2** does not stem from the syntax of the formula, but from the interpretation of $\neg \lhd \neg\bot$ and its analog in the other direction. These constructs pick out the edges of the string, but it is not readily apparent how they do so. First recall that $\bot$ is false at every node in the model. So the negation of $\bot$ is true at every node. Now $\lhd \neg\bot$ is true at a node w if and only if there is a node immediately to the left of w where $\neg\bot$ is true. Clearly this will be true no matter what the node will look like. However, if there is no such node to begin with, i.e. if w is the leftmost node in the string, then the predicate will be false because the $\lhd$-operator presupposes the existence of a node to the left of w. From this simple line of reasoning it follows that $\neg \lhd \neg\bot$ is satisfied only at the left edge of the string. The case with $\rhd$ works exactly the same.

Proceeding with the formalization, we turn to GP's feature calculus. Here we need to introduce some further terminology. It is worth pointing out that the terminology is not part of the logic itself but belongs only to the metalanguage used in its description. A propositional formula ϕ over a set of variables $x_1, \ldots, x_k$ is called *exhaustive* iff $\phi := \bigwedge_{1 \leq i \leq k} \psi_i$, where for every i, ψ_i is either x_i or $\neg x_i$. This means that exhaustive formulas do not allow for underspecification (and indeed we use them to describe PEs, which are fully specified with respect to their feature makeup). A PE ϕ is an exhaustive propositional formula over $\mathscr{E}$ such that $\{\phi, \mathbf{F1}, \mathbf{F2}, \mathbf{F3}, \mathbf{F4}, \bigvee h, \bigvee o\}$ is consistent (i.e. none of the formulas contradict each other).

F1	$\bigwedge(h_n \rightarrow \bigwedge_{h_n \neq h'_n} \neg h'_n)$	Exactly one head
F2	$\neg \underline{v} \rightarrow \bigwedge(h_n \rightarrow \bigwedge_{\pi_1(h)=\pi_1(o)} \neg o_n)$	No basic element (except v) twice
F3	$\overline{v} \rightarrow \bigwedge_{o \neq \overline{v}} \neg o$	$\overline{v}$ excludes other operators
F4	$\bigwedge(e_2 \rightarrow \bigvee h_1 \wedge \bigvee o_1)$	Pseudo branching implies first branch

Let PH be the least set containing all PEs (noting that a PE is now a particular kind of propositional formula), and let $lic : PH \to \wp(PH)$ map every PE to its set of melodic licensors. Furthermore, $S \subseteq PH$ designates the set of PEs that may occur in the codas of magic licensing configurations (the letter S is mnemonic for "sibilants"). Once again neither *lic* nor S are part of the logic itself. The following five axioms, then, sufficiently restrict the melody.

M1	$\bigwedge_{i \in \mathscr{S}} \left(i \to \bigvee_{\phi \in PH} \phi \vee \mu \vee \texttt{fake} \right)$	Universal annotation
M2	$((O \vee C \vee \lhd N \vee \rhd N) \to \bigwedge \neg e_2)$	No pseudo branching for O, C & branching N
M3	$O \wedge \lhd O \to \bigwedge_{\phi \in PH} (\phi \to \bigvee_{\psi \in lic(\phi)} \lhd \psi)$	Licensing within branching onsets
M4	$C \wedge \bigwedge_{i \in S} \neg i \to \lhd \neg \mu \wedge \bigwedge_{\phi \in PH} (\phi \to \bigvee_{\psi \in lic(\phi)} \rhd \psi)$	Melodic coda licensing
M5	$\texttt{fake} \to O \wedge \bigwedge_{m \neq \texttt{fake}} \neg m$	Fake onsets

For the sake of illustration, let us take a closer look at **M4**. The antecedent of the conditional, $C \wedge \bigwedge_{i \in S} \neg i$ can be translated as "you are a coda that does not host a sibilant". The consequent then enforces two well-formedness conditions. First, the coda is preceded by a pronounced node ($\lhd \neg \mu$; this is guaranteed to be a nucleus due to the peculiarities of the syllable template, so we do not have to mention this fact explicitly). The second condition is more complicated. The big connective notation should be sufficiently familiar by now. The new twist is the invocation of the function *lic* in the subscripts. The idea is that for every PE ϕ, *lic* picks out its licensors such that the variable ψ can be instantiated accordingly. So if, for instance, there are exactly two PEs ϕ and ϕ', and both of them have

only the licensors described by the formulas ψ and ρ, the full extension of **M4** is $(\phi \rightarrow \triangleright\, \psi \vee \triangleright\, \rho) \wedge (\phi' \rightarrow \triangleright\, \psi \vee \triangleright\, \rho)$.

Remember that GP allows languages to impose further restrictions on the melody by recourse to licensing constraints. It is easy to see that licensing constraints operating on single PEs can be captured by propositional formulas. The licensing constraint "A must be head", for instance, corresponds to the propositional formula $\neg\overline{A}$. Licensing constraints that extend beyond a single segment can be modeled using $\triangleleft$ and $\triangleright$, provided their domain of application is finitely bounded (see the discussion on spreading below for further details). Thus licensing constraints pose no obstacle to formalization in our logic, either.

With skeletal and melodic constraints taken care of, we can turn to the two most intricate modules of GP, empty categories and spreading, starting with the former. As mentioned above, I use μ to mark "mute" segments that will be realized as the empty string. The distribution of μ is very simple for O and C — C never allows it, and O only if it is unary branching and followed by a pronounced N. For N, on the other hand, we first need to distribute $\checkmark$ in a principled manner across the string to mark the licensed nuclei, i.e. those N that may remain unpronounced. Note that unpronounced segments must not contain any other elements (which would affect spreading).[7]

L1	$\mu \rightarrow \bigwedge_{m \notin \{\mu, \checkmark\}} \neg m \wedge \neg C \wedge (N \rightarrow \checkmark)$	Empty categories
L2	$N \wedge \triangleleft N \rightarrow (\mu \leftrightarrow \triangleleft \mu)$	No partially mute branching nuclei
L3	$O \wedge \mu \rightarrow \neg \triangleleft O \wedge \triangleright (N \wedge \neg\mu)$	Mute onsets

[7] The constraint can be relaxed such that unpronounced segments must not contain any local elements but may perfectly well host spread elements, in which case they would act as an invisible landing site for feature spreading to overcome locality barriers. This slight modification is incorporated by changing the subscript of the big connective in **L1** to $\pi_1(m) = \mathit{local}$.

L4 $N \wedge \checkmark \leftrightarrow \underbrace{\rhd (C \wedge \bigvee_{i \in S} i)}_{\text{Magic Licensing}} \vee \underbrace{(\neg \lhd N \wedge \neg \rhd \neg \bot)}_{\text{FEN}} \vee \underbrace{((\neg \lhd N \rightarrow \lhd (\lhd N \vee \neg \lhd \neg \bot)) \wedge (\neg \rhd N \rightarrow \rhd \rhd (N \wedge \neg \mu)))}_{\text{Proper Government}}$ P-licensing

Axiom **L4** looks intimidating at first but is easy to unravel. It states that a nucleus is licensed iff it satisfies at least one of three conditions: Magic Licensing, FEN or Proper Government. The Magic Licensing conditions is simply a logical description of the configurations that allow for Magic Licensing: N is licensed if it is followed by a sibilant in coda position.[8] The FEN condition ensures that wordfinal N are licensed if they are non-branching (recall that $\neg \rhd \neg \bot$ picks out the right word edge). The proper government condition is the most complex one, though it is actually simpler than the original GP definition, as I was eager to point out in the preceding section. As I explained there, N is properly governed if the first N following it is pronounced and neither a branching onset nor a coda intervenes. Since we treat a binary branching constituent as two adjacent unary branching constituents, the proper government condition can be reinterpreted as a structural requirement such that N (or the first N if we are talking about two adjacent N) may not be preceded by two constituents that are not N and (the second N) may not be followed by two constituents that are not N or not pronounced. Together with axioms **S1**–**S7**, this gives the same results as the original constraint.[9]

[8]Note that we can easily restrict the context, if this appears to be necessary for empirical reasons. Strengthening the condition to $\rhd (C \wedge \bigvee_{i \in S} i) \wedge \lhd \neg \lhd \neg \bot$, for example, restricts magic licensing to the N occupying the second position in the string.

[9]In this case, the modal logic is once again flexible enough to accommodate various alternatives. For instance, if proper government should be limited to non-branching Ns, one only has to replace both occurrences of $\rightarrow$ by $\wedge$. Also, my formalization establishes no requirement for a segment to remain silent, because N often are pronounced in magic licensing configurations or at the end of a word in a FEN language. For proper government, however, it is sometimes assumed that licensed nuclei have to remain silent, giving rise to a strictly alternating pattern of realized and unrealized Ns. If we seek to accommodate such a system, we have to distinguish Ns that are magically licensed or FEN licensed from Ns that are licensed by virtue of being properly governed. The easiest way to do so is to split $\checkmark$ into two features $\checkmark_o$ and $\checkmark_m$ (optional and mandatory), the latter of which

Before we move on to spreading, it should be pointed out how simple our axioms are from a logical perspective. An easy way of measuring the complexity of modal formulas is to look at their *quantifier depth*, which is a measure of how many levels of modal operator nesting we find.[10] It is defined as follows, where $qd(\phi)$ denotes the quantifier depth of ϕ:

- If p is a proposition symbol, $qd(p) = 0$.
- If ϕ is a formula, $qd(\neg\phi) = qd(\phi)$.
- If ϕ and ψ are formulas and $\cdot \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, $qd(\phi \cdot \psi) = max(qd(\phi), qd(\psi))$ (i.e. the greater of the two).
- If ϕ is a formula and $\langle o \rangle$ some modal operator, $qd(\langle o \rangle \phi) = qd(\phi) + 1$.

So $qd(p \wedge \lhd q) = qd(p \wedge \rhd q) = 1$, but also $qd(\lhd (\lhd p \wedge q)) = qd(\lhd (\lhd p \wedge \lhd q)) = 2$, even though the latter contains more modal operators. Most intriguingly, almost all the axioms above have a quantifier depth of 1 or less. The only exception is Proper Government, which has a quantifier depth of 2 (this is surprising considering the complexity of the original definitions and might be taken to highlight the underlying simplicity of the idea). Obviously these are very small numbers — even very simple modal formulas reach high quantifier depth values very fast — which suggests that GP is indeed a highly restricted theory.

is reserved for properly governed Ns. The simple formula $\checkmark_m \rightarrow \mu$ will force such Ns to remain unpronounced.

[10] This measure is commonly used for classical logic, i.e. variants of first-order logic. Recall for instance from Fn. 3 that each quantifier in an MSO formula may induce an exponential blow-up in the number of states of the corresponding automaton. It is also known that a first-order formula with n quantifiers can tell whether two strings are of the same length only if at least one of them is of length $l < 2^n$ (Libkin 2004), whence a formula with more quantifiers can distinguish more strings. Examples of the importance of quantifier depth for modal logics are too technical to explain here, but do exist (Halpern 1995; Infante-Lopez et al. 2003).

Quantifier depth will increase significantly, however, with the introduction of spreading. Or rather, it might increase significantly, as most properties of spreading are language specific — only the set of spreadable features and the ban against onset internal spreading are universal. To capture this variability, I define a general spreading scheme σ with six parameters i, j, ω, $\mathfrak{o}$, *min* and *max*. Since this scheme is already supremely confusing to read as it is, I omit for now the ban against onset internal spreading.

$$\sigma := \bigwedge_{\pi_1(i)=\pi_1(j)} (i \wedge \omega \rightarrow \bigvee_{n=min}^{max} \Diamond^n (j \wedge \mathfrak{o}))$$

As I just mentioned, this is not a true formula but rather a formula scheme, that is to say, a parameterized template from which we can generate the intended formulas by substituting the desired values for the parameters. Here the variables $i, j \in \mathscr{E}$, coupled with judicious use of the formulas ω and $\mathfrak{o}$ regulate the optionality of spreading. If spreading is optional, i is a spread element and ω, $\mathfrak{o}$ are formulas describing, respectively, the structural configuration of the target of spreading and the set of licit sources for spreading operations to said target. If spreading is mandatory, then i is a local element and ω, $\mathfrak{o}$ describe the source and the set of targets. If we want spreading to be mandatory in only those cases where a target is actually available, ω has to contain the subformula $\bigvee_{n=min}^{max} \Diamond^n \mathfrak{o}$. Observe moreover that we need to make sure that every structural configuration is covered by some ω, so that unwanted spreading can be blocked by rendering $\mathfrak{o}$ not satisfiable. As further parameters, the finite values $min, max > 0$ encode the minimum and maximum distance of spreading, respectively. Finally, the operator $\Diamond \in \{\triangleleft, \triangleright\}$ fixes the direction of spreading for the entire formula ($\Diamond^n$ is the n-fold iteration of $\Diamond$). With optional spreading, the direction of the operator is opposite to the direction of spreading, otherwise they are identical. The different ways of interaction between

Mode	Direction	i	ω	ϖ	$\Diamond$
optional	left	spread	target	source	$\triangleright$
optional	right	spread	target	source	$\triangleleft$
mandatory	left	local	source	target	$\triangleleft$
mandatory	right	local	source	target	$\triangleright$

Table 2.1: Parameterization of spreading patterns with respect to σ

the parameters is summarized in Table 2.1.

To incorporate the ban against onset internal spreading, we only have to increase the minimum spreading distance by one for branching onsets. This yields the extended scheme below.

$$\sigma := \bigwedge_{\pi_1(i)=\pi_1(j)} (i \wedge \omega \rightarrow \bigvee_{n=min}^{max} \Diamond^n(j \wedge \varpi) \wedge (O \wedge \Diamond O \rightarrow \bigvee_{n=min+1}^{max} \Diamond^n(j \wedge \varpi)))$$

As the astute reader (and all readers that glimpsed at footnotes 7, 8 and 9) will have noticed by now, nothing in our logic prevents us from defining alternative versions of GP. Whether this is a welcome state of affairs is a matter of perspective. On the one hand, the flexibility of our logic ensures its applicability to a wide range of different variants of GP, e.g. to versions where spreading is allowed within onsets or where the details of proper government and the restrictions on branching vary. On the other hand, it raises the question whether there isn't an even weaker modal logic that is still expressive enough to formalize GP. However, the basic feature calculus of GP already requires the logical symbols $\neg$ and $\wedge$, which gives us the complete set of logical connectives, and we furthermore need $\triangleleft$ and $\triangleright$ to move us along the phonological string. Hence, imposing any further syntactic restrictions on formulas requires advanced technical concepts such as the number of quantifier alternations. But this brings us back to an issue I discussed in the preface to this section: the loose grip of mathematical methods, and why it isn't as problematic as

it might seem initially. Lest I unnecessarily bore the reader with methodological remarks, I shall merely point out that it is doubtful that a further weakening of the logic would have interesting ramifications given the questions I set out to answer; I am not interested in the logic that provides the best fit for a specific theory but in the investigation of entire classes of string-based phonological theories from a model-theoretic perspective. In the next section, I try to get closer to this goal.

CHAPTER 3

Formal Comparison of Theories

3.1 Overview

With a formal model of GP in place, it is time to see if my claims about MTP — foremost that it provides us with a general perspective on the expressivity of phonological theories and factors out the components that enhance or restrict it — were overly optimistic. In order to prove that they were not, I show why spreading rather than p-licensing, the syllable template or the feature system can be viewed as the locus of power in GP. I first demonstrate how spreading can be generalized in a very natural way by increasing the power of our modal logic, and I conduct empirical case studies to determine if this power is actually needed, thereby buttressing further claims of mine that MTP has something to offer for strictly empirically-minded linguists, too. After feature spreading has been established as a decisive factor for the generative capacity of a theory, I show that the same is not true of melodic and skeletal restrictions.

3.2 The Phonological Hierarchy

Implicit in the closing discussion of the previous chapter was the insight that our logic is powerful enough to account for all finitely bounded phonological phenomena. The reason is that if a phenomenon, say assimilation between onsets,

is restricted to a domain with an upper limit on its size, e.g. to onsets that are separated by exactly one rhyme, then there are only finitely many configurations that have to be regulated by our logic. In the case at hand, we only have to consider substrings of the form onset-rhyme-onset, and since the length of onsets and rhymes is itself bounded to a maximum of 2, we only have to consider strings of length 6 or less. This means that a formula of finite quantifier depth is sufficient to enforce the desired restrictions. In linguistic parlance, we might also say that all local processes, rules and constraints — including many unattested ones — can be accounted for in our logic (note that this does not imply that GP itself can account for all of them, since many of them will be ruled out by, say, the syllable template or the ECP).

Unbounded processes, i.e. long-distance phenomena that allow an unlimited amount of phonological material to intervene between source and target, are more problematic. Our logic above is limited to formulas of finite length, whence each formula can only see a certain number of steps to the left or the right. Its viewing distance is determined by its quantifier depth. For example, a formula like $\lhd p \vee \rhd\rhd q$ sees one step to the left and two steps to the right. At an abstract level, then, we might associate with a phonological theory a window that moves over the phonological string and shows us only n many consecutive symbols at a time, where n is the highest quantifier depth value of all the formulas encoding the theory. This perspective reinforces at an intuitive level the claim above that every strictly local phenomenon can be modelled by this logic, and it also makes it obvious why unbounded processes pose a challenge to such a "moving window" version of phonology.

Interestingly, though, it is possible to reconcile long-distance phenomena with the moving window model, provided that they can be reinterpreted as arising from iterated application of finitely bounded processes or conditions. Consider for

example a stress rule for some language L that assigns primary stress to the last syllable that is preceded by an even number of syllables. Assume furthermore that secondary stress in L is trochaic, that is to say it falls on every odd syllable but the last one. Let 1 and 2 stand for primary and secondary stress, respectively. Unstressed syllables are assigned the feature 0. Then the following formula will ensure the correct assignment of primary stress (for the sake of simplicity, we assume that every node in the string represents a syllable; it is an easy but unenlightening exercise to rewrite the formula for our GP syllable template).

$$\bigvee_{i\in\{0,1,2\}} i \wedge \bigwedge_{i\neq j\in\{0,1,2\}} (i \rightarrow \neg j) \wedge (\neg \lhd \neg\bot \rightarrow 1 \vee 2) \wedge (2 \rightarrow\rhd 0)\wedge$$
$$(0 \rightarrow\rhd (1 \vee 2) \vee \neg \rhd \neg\bot) \wedge (1 \rightarrow \neg \lhd 1 \wedge (\neg \rhd \neg\bot \vee \rhd \neg \rhd \neg\bot))$$

The intuition here is that we do not need a window that covers the entire string in order to determine if we are an odd or an even number of syllables away from the left edge of the string because this information can be recovered in a local way from the position of secondary stress, which is itself assigned in a strictly local manner. Other seemingly unbounded phenomena arising from iteration of local processes, most importantly vowel harmony (see Charette and Göksel 1994, 1996 for a GP analysis), can be captured in a similar way. However, *a priori* we cannot rule out the existence of unbounded phonological phenomena that require increased expressivity. The lingering question, then, is how our logic can be modified accordingly.

Depending on the mathematical background of the reader, it might come as no surprise that such modifications are readily available and have been studied extensively in the literature, in particular in the form of various incarnations of linear temporal logic (see Kröger and Merz 2008 for a thorough introduction). The name might cause unnecessary confusion, but the reader can rest assured that

temporal logics differ from the modal logics we have encountered so far only in the addition of new operators that significantly increase their power.

The first natural step in enhancing our modal logic is the addition of two operators $\triangleleft^+$ and $\triangleright^+$ with the corresponding relation $R^+_\triangleleft$, the transitive closure of $R_\triangleleft$ (recall from Sec. 2.3 that operators need a relational counterpart defined over the frame). A node n_1 is related to n_m by $R^+_\triangleleft$ iff there is are nodes n_2, …, n_{m-1} such that $R_\triangleleft$ relates n_1 to n_2, n_2 to n_3 and so on. In other words, the operators $\triangleleft^+$ and $\triangleright^+$ simply mean "somewhere to the left" and "somewhere to the right", respectively, so their field of view is unbounded. They come with a significant disadvantage though: They have no depth perception. The relation $R^+_\triangleleft$ does not keep track of how many steps to the left or to the right one has to take, it just tells us that there is some sequence of left/right-successors that will eventually take us where we want to go. Thus, in combination with the $\triangleleft$ and $\triangleright$, what we get in terms of locality might be viewed as a window of finite size (determined by the depth of $\triangleleft$ and $\triangleright$ nestings), beyond the edges of which we only see a hazy cloud of nodes without being able to gauge which are closer to us and which farther away.

Mathematically inclined readers might be delighted to hear that this new logic is exactly as powerful as restricted temporal logic (Cohen et al. 1993), which in turn has been shown in Etessami et al. (1997) to exactly match the expressivity of the two-variable fragment of first-order logic (FO^2; see Weil 2004 for further equivalence results). Linguists, on the other hand, will appreciate that among other things, OCP effects (Leben 1973; Goldsmith 1976) can now be captured in an elegant way. The formula $O \wedge \underline{\mathbf{A}} \wedge \overline{\mathbf{L}} \wedge \overline{?} \rightarrow \triangleright^+ \neg(O \wedge \underline{\mathbf{A}} \wedge \overline{?})$, for example, disallows alveolar nasals to be followed by another alveolar stop, no matter how far the two are apart. Note that this entails that if we fail to restrict the domain of the formula, the OCP effect may hold even across word boundaries, which isn't desirable in the

general case.

Due to their depth perception problem, $\lhd^+$ and $\rhd^+$ are too coarse for faithful renditions of unbounded spreading. They are incapable of defining, say, all intervals of arbitrary size within which a certain condition has to hold (e.g. no b may appear between a and c, or a suitably restricted variant of the OCP formula above). This should be evident once one realizes that in order to restrict an interval, one has to locate the start point, the end point, and the nodes in between. If one doesn't know the linear order between the nodes, this is impossible to do.

An easy remedy comes in the form of the until and since operators U and S that can be considered a trademark of the canonical version of linear temporal logic. Both are dyadic operators, which means that they scope over two formulas rather than one. So instead of $U\phi$, one writes $U(\phi, \psi)$. The interpretation of this formula, in intuitive terms is, "somewhere to the left of here, there is a node n at which ϕ is true such that at no node between here and n, ψ is true". For instance, $O \rightarrow U(\underline{\mathbf{A}}, \neg\underline{\mathbf{I}} \wedge \neg\bar{\mathbf{I}})$ asserts that for every onset o, there is a node n somewhere to the left of o with element **A** in head position and no **I** occurring between o and n, neither in head nor in operator position. Since the notion of an interval is directly built into U and S, they can easily step in for $\lhd^+$ and $\rhd^+$ should intervals be needed.

As for their mathematical properties, we know that the addition of these operators grants us the power of full first-order logic, which, somewhat perplexingly, is identical to the power of the three-variable fragment of first-order logic (FO^3) when we restrict our attention to strings (Immerman and Kozen 1989). This is to say, the minimal increase in expressivity from FO^2 to FO^3 affords us all the power we need to talk about intervals. From the equivalence with first-order logic it also follows that we can now define even star-free languages (McNaughton and Pappert 1971; Thomas 1979; Cohen 1991; Cohen et al. 1993). The star-free languages form

the most powerful class of string languages that are still strictly weaker than the regular languages.

To go the last mile and push the logic to the level of regular languages, we could add a rather complicated device called fixed point operators (Vardi 1988), or alternatively, directly incorporate regular expressions as they are commonly used in computer science for representing regular languages (Leucker and Sánchez 2005). In either case, the syntax and semantics of the logic get rather complicated, whence even short examples would pose the risk of derailing the discussion. For practical purposes, it would probably be easier to use another equivalent logic instead, monadic second-order logic (Büchi 1960), but these issues are peripheral to our discussion. The general upshot that all three extensions enable us to define more elaborate spreading patterns that are unbounded, can recognize intervals, and even carry out simple computations like counting *modulo* m to the right starting from some node n (which will pick out all the nodes that are m, $2m$, $3m$, ... steps to the right of n; if $m = 2$, this is equivalent to distinguishing odd from even nodes). Still we cannot capture more elaborate patterns that are very common in syntax, foremost unbounded center embedding and unbounded cross-serial dependencies. It is generally assumed, though, that these have no role to play in phonology (see also my brief remarks at the very end of Sec. 1.2).

Table 3.1 on the next page summarizes the properties of the extensions of the basic GP logic that were surveyed in this section. Recall from Sec. 1.2 that SPE defines exactly the class of regular languages (Johnson 1972; Kaplan and Kay 1994). Hence, the class of GP theories with elaborate spreading patterns is identical to the class of SPE theories.

	$GP^{\triangleleft}$	$GP^{\triangleleft^+}$	GP^{U}	GP^{ν}/SPE
Modal logic	$\triangleleft$	RTL	LTL	ν-LTL = RLTL
Predicate logic	—	FO^2	$FO^3 = FO$	MSO
Formal language	—	—	star-free	regular

Table 3.1: Hierarchy of classes of phonological theories

3.3 How Much Expressivity is Needed?

3.3.1 Caveat: The Power of Feature Coding

In this section, I show that two attested phonological phenomena require a more powerful formalism than standard GP. The result needs to be prefaced by a short disclaimer, though. Invoking a comparatively little-known mathematical theorem (Thatcher 1967), one can show that all variants of GP can define regular languages, the highest level of our hierarchy, if non-local dependencies may be encoded by diacritic features, as this allows us to reinterpret unbounded dependencies as the result of iterated local spreading of these diacritic features. Painting in broad strokes, a feature is a diacritic if it never has a visible effect on the surface string, such as the slash-feature of GPSG in syntax (Gazdar et al. 1985).[1] To recapitulate in linguistic terms, Thatcher's result tells us that for any SPE grammar over a set $\mathscr{F}$ of features, there is an equivalent GP grammar over a set of features that is an extension of $\mathscr{F}$. But fortunately, the use of such diacritic features is frowned upon by linguists, and if we assume that the set of features is fixed across all theories, the expressivity hierarchy above holds unchanged. Nevertheless, the power of feature coding forces us to explicitly relativize the results to specific feature sets, which

[1] The notion is difficult to pin down because if we assume a universal feature set, certain features might be redundant in language L but not in language L', whence they would not be a diacritic in the strict sense but could still be used in this way for language L. I expect that in those borderline cases, one would also find disagreement between linguists whether an analysis (ab)using those redundant features would be considered insightful.

makes them somewhat cumbersome to read.

3.3.2 Beyond $GP^{\triangleleft^+}$ — Sanskrit n-Retroflexion

My first case study revolves around a well-known long-distance phenomenon, n-retroflexion in Vedic Sanskrit, also known as *nati*. As discussed in Schein and Steriade (1986) and Hansson (2001) (building on data given in Whitney 1889 and Macdonell 1910), the process turns the first n following a continuant retroflex consonant (r, ʂ) into a retroflex ɳ.

As shown in (10), the process applies under direct adjacency, across vowels, and across (non-coronal) consonants. Notably, the process is apparently non-local, i.e. there is no upper bound on the length of the intervening material (see Whitney 1889:225–230 for the full data set).

(10) a. /iʂ-naː/ → iʂ-ɳa-

b. /cakʂ-aːna-/ → cakʂ-aɳa

c. /vrk-na-/ → vr̥k-ɳa-

d. /krp-a-maːna-/ → kr̥p-a-maːɳa-

That *nati* is indeed non-iterative, that is to say, it does not apply to any n after the first one, is witnessed by (11).

(11) a. pra-ɳi̯naːya (*pra-ɳi-ɳaːya)

b. kr̥ɳ-vaːna (*kr̥ɳ-vaːɳa)

In (12), we see that the process is blocked by any intervening coronal consonants (dental, palatal, as well as retroflex ones, and both obstruents and sonorants).

(12) a. mr̥d-naː- (*mr̥d-ɳaː-)

b. marj-aːna- (*marj-aːɳa-)

c. kr̥t-a-maːna- (*kr̥t-a-maːɳa-)

d. kʂved-aːna- (*kʂved-aːɳa-)

Moreover, *nati* is blocked if the target is not immediately followed by a sonorant.

(13) a. brahman (*brahmaɳ)

b. tr=n=t-te (*tr=ɳ=t-te)

Finally, the target may not be followed by a segment that could itself be followed by a source for *nati*, i.e. n may not be followed by another continuant retroflex consonant.

(14) a. praː-nr̥tyat (*praː-ɳr̥tyat)

b. pari-nakʂati (*pari-ɳakʂati)

From the data we derive the following description: *nati* turns the first n following a continuant retroflex consonant (r, ʂ) into a retroflex ɳ iff the following conditions are fulfilled:

(15) a. No coronal consonant intervenes between trigger and target.

b. The nasal is immediately followed by a (nonliquid) sonorant.

c. The nasal is not followed by a retroflex continuant.

We now prove in two steps that a process like *nati* cannot be captured by the class $GP^{\triangleleft}$ or $GP^{\triangleleft^+}$. The first theorem asserts that there is a GP-variant that is expressive enough to faithfully model this process, whereas the second demonstrates the insufficiency of $GP^{\triangleleft}$ and $GP^{\triangleleft^+}$.

Theorem 3.1. *There is a feature set $\mathscr{F}$ such that there is a theory over $\mathscr{F}$ that belongs to the class GP^U and accounts for nati.*

Proof. The conditions in (15) translate into the three GP^U axioms below. Remember that the feature μ is used to denote unpronounced segments. Furthermore, for any sound or class of sounds i, we use $\ulcorner i \urcorner$ to denote the PE i, i.e. the propositional formula over $\mathscr{F}$ that uniquely represents i. In the special case of $\ulcorner$*derived* $\eta \urcorner$, this corresponds to a formula that is obtained from the formula $\ulcorner \eta \urcorner$ by replacing the feature(s) that distinguish η from n by their spread analogue. It is easy to see that there are feature systems where η has more features than n (so that η has some features that can be replaced) and where all expressions in the axioms can be uniquely represented. Hence there is at least one suitable $\mathscr{F}$. Together with the three GP^U axioms this proves the theorem.[2]

N1 $\ulcorner$*derived* $\eta \urcorner \rightarrow U(\ulcorner r \urcorner \vee \ulcorner \text{ʂ} \urcorner, \neg \ulcorner$*coronal*$\urcorner \wedge \neg \ulcorner n \urcorner)$

"If *derived* η is true at node w, then there is a node v labeled r or s to the left of w and neither a coronal nor an n occurs between v and w."

N2 $\ulcorner$*derived* $\eta \urcorner \rightarrow \triangleright \ulcorner$*sonorant*$\urcorner$

"If *derived* η is true at node w, then there is a sonorant immediately to the right of w."

N3 $\ulcorner$*derived* $\eta \urcorner \rightarrow \neg \triangleright^{+} \ulcorner$*retroflex continuant*$\urcorner$

"If *derived* η is true at node w, then there is no retroflex continuant to the right of w." □

[2] If there are other processes $P_1, \ldots, P_n$ that also derive η, their are two options. Either we say that *nati* is a last resort operation, so it applies only if no other process is applicable. In this case, the antecedent of each axiom has to be strengthened to $\ulcorner$*derived* $\eta \urcorner \wedge \bigwedge_{i=1}^{n} \neg \phi_i$, where each ϕ_i is a description of the configuration under which P_i may take place. Alternatively, we may say that a surface string is licit if it can be derived by some process, in which case the axioms have to be combined into one as follows:

$$\ulcorner \textit{derived } \eta \urcorner \rightarrow (U(\ulcorner \mathrm{r} \urcorner \vee \ulcorner \text{ʂ} \urcorner, \neg \ulcorner \textit{coronal} \urcorner \wedge \neg \ulcorner \mathrm{n} \urcorner) \wedge \triangleright \ulcorner \textit{sonorant} \urcorner \wedge \neg \triangleright^{+} \ulcorner \textit{retroflex continuant} \urcorner) \vee \bigvee_{i=1}^{n} \phi_i$$

Since we are only interested in the complexity of *nati* itself, we disregard these special cases.

For the second theorem, though, an additional assumption has to be introduced, namely that we have to account for nati at the sentence-level, rather than the word-level. That is to say, our phonological theory has to decide for entire sentences whether they are well-formed, rather than for words in isolation. This assumption is actually rather plausible, as there are plenty of phonological processes that operate across word boundaries yet do not seem to behave any different from word-internal processes. We have to introduce this additional assumption because nati is in fact $GP^{\lhd^+}$-definable if we restrict our attention to isolated words. This is due to **N3**, which requires that the target of nati may not be followed by any sounds that could act as nati-triggers. With this restriction in place, nati can be captured by the following formula:

$$\ulcorner \textit{derived } \mathrm{n}\urcorner \rightarrow \neg \lhd^+ (\ulcorner \textit{retroflex continuant}\urcorner \wedge \neg \rhd^+ (\ulcorner \mathrm{r}\urcorner \vee \ulcorner \mathrm{s}\urcorner))$$

It states that if there is a retroflex continuant somewhere to the left of a nati target, then somewhere to the right of said continuant, there has to be a nati trigger — as **N3** does not allow the trigger to be to the right of the target, it has to be between the target and the continuant. In particular, this entails that no retroflex continuant occurs between the target of the nati and the first source to its left. Now if we consider sentences rather than isolated words, **N3** does not ban the occurence of triggers to the right of targets anymore and the formula above does not have the intended result.[3]

Keeping this important caveat in mind, we may now turn to the second theorem.

[3] A pathological but nevertheless noteworthy special scenario arises if our feature system cannot distinguish between lexical and derived instances of n. As n may also be phonemic in Sanskrit (Kobayashi 2004), the formula above would give the desired result only if the distribution of phonemic n can be described in terms of $GP^{\lhd^+}$-formulas. This ties in with my remarks in Fn. 2. Thanks to Craig Melchert for bringing the dual status of n to my attention and pointing me to the relevant literature.

Theorem 3.2. *There is a feature set $\mathscr{F}$ such that at least one theory over $\mathscr{F}$ which belongs to the class GP^U accounts for nati at the sentence-level but no theory over $\mathscr{F}$ belonging to a weaker class does.*

Proof. Since axioms **N2** and **N3**, which represent (15b) and (15c), are in fact $GP^{\triangleleft^+}$ formulas, the culprit has to be **N1**, the formal encoding of (15a). As we have previously seen, $GP^{\triangleleft^+}$ is a special variant of restricted temporal logic over strings, which can be formalized in FO^2. Hence it suffices to show that (15a) cannot be stated in FO^2. This can be demonstrated using standard techniques from finite model theory. Due to its formal difficulties, though, the proof is relegated to the appendix. But even on an intuitive level it should be manifest that we need three variables for (15a): two in order to mark the edges of the interval defined by trigger and target, and a third one to restrict the nodes within said interval. □

Presumably the astute reader did already realize that the proof of Thm. 3.2 establishes a stronger result: given such a $\mathscr{F}$, there is no theory in the class $GP^{\triangleleft^+}$ that accounts for *any* process which involves checking nodes within an interval of unbounded size. Under the proviso that there are empirical phenomena besides *nati* that exhibit this property, Thm. 3.2 immediately gives us a catalog of phenomena that $GP^{\triangleleft^+}$-theories cannot account for without careful tweaking of their feature system.

But the discovery of further "unbounded interval" processes is likely to prove difficult, because it is a notoriously hard task to establish conclusively that the unboundedness of the size of the interval doesn't arise from the iteration of bounded spreading steps. In the case of *nati*, for instance, it is also conceivable (see Hansson 2001:242f) that we are actually dealing with a sequence of local retroflexion-steps only the last of which is marked in the Sanskrit writing system. Then *nati* could easily be accommodated in a $GP^{\triangleleft}$-theory. Well-formedness conditions not involving

any spreading to begin with (e.g. "no s between p and t") would thus constitute a better place to look for such unbounded intervals, but they all seem to be restricted to small phonological constituents such as syllables or onsets, for the size of which we can establish an upper bound.

3.3.3 Beyond GP^U — Primary Stress Assignment in Creek and Cairene Arabic

We now turn to primary stress assignment in Creek and Cairene Arabic (Mitchell 1960; Haas 1977; McCarthy 1979). I only list the stress rules of Cairene Arabic, as our general reasoning carries over to Creek just as well.

(16) *Stress assignment in Cairene Arabic*

a. Stress the final syllable, if it is superheavy (CV:C or CVCC).

b. Else stress the penult, if it is heavy (CV: or CVC).

c. Else stress the penult or the antepenult, whichever is separated by an even number of syllables from the closest preceding heavy syllable (or, if there is no such syllable, from the beginning of the word).

d. There is no overt marking of secondary stress.

Ignoring for a moment (16d), note that conditions (16a) – (16c) are fairly unremarkable from a typological perspective and can be found in hundreds of languages, often in conjunction with trochaic or iambic secondary stress assignment. In fact, we have already encountered a simplified variant of such systems in Sec. 3.2 as an example of how unbounded dependencies can arise from the iteration of local dependencies. So we already know that such a system can be captured in $GP^{\lhd}$, and hence in GP^U, too.[4]

[4]Recall that the technical details are slightly more intricate than is apparent due to stress rules operating on syllables, which do not exist as discrete entities in GP.

The important insight in Sec. 3.2 was that the distribution of primary stress can be computed in a strictly local manner from the position of secondary stress, which can be assigned in a local fashion, too. This approach is much in the spirit of metric analyses with binary branching feet (Hayes 1995). In the case of Cairene Arabic, however, we cannot rely on secondary stress: (16d) makes it clear that there is no overt secondary stress in Cairene Arabic, and as a consequence, the secondary stress feature is degraded to the status of a coding feature, and as such it cannot be assumed to be an integral part of all feature systems.[5] As a consequence, $GP^{\triangleleft}$, or as we will see in a second, all variants up to GP^{U} are too weak to define the correct constraints on primary stress assignment.

Theorem 3.3. *There is a feature set $\mathscr{F}$ such that there is a theory over $\mathscr{F}$ that belongs to GP^{v} and accounts for primary stress assignment in Cairene Arabic.*

Proof. From our discussion above it follows that the crucial factor in assigning primary stress is counting the number of syllables. In particular, we need to be able to distinguish even from odd syllables. This corresponds to counting *modulo* 2 (remember that *n modulo* 2 is the remainder of dividing *n* by 2, so *n modulo* 2 is 0 if *n* is even and 1 otherwise). We know from the combined work of Büchi (1960) and Vardi (1988) that a language can be defined in GP^{v} iff it is regular. There is also a simple and well-known algorithm for constructing a finite state automaton that counts *modulo n*, *n* finite, whence counting *modulo n* does not exceed the power of regular languages. *A fortiori*, then, any GP^{v} theory can count *modulo 2*. Note that since the only feature needed for assigning stress is a primary stress feature, almost every choice of $\mathscr{F}$ is sufficient. □

[5] This should not be construed as a claim about the validity of metric approaches such as Hayes's. It is merely a strict methodological requirement that we have to enforce here to prevent the phonological hierarchy from collapsing and ensure the general validity of our theorems. It might well be the case that secondary stress is computed even though it has no overt exponent. In fact, the loose mapping from phonology to phonetics assumed in GP makes this a very plausible assumption.

Theorem 3.4. *There is a feature set $\mathscr{F}$ such that at least one theory over $\mathscr{F}$ which belongs to the class GP^{V} accounts for nati but no theory over $\mathscr{F}$ belonging to a weaker class does.*

Proof. It is well known that full first-order logic cannot count *modulo n*. As was proved by McNaughton and Pappert (1971) and Thomas (1979), the stringsets definable in first-order logic are the star-free stringsets, which in turn are also the stringsets definable in LTL (Cohen et al. 1993), in which we formalized GP^{U}. □

Once more I feel obliged to point out that my line of reasoning here hinges on data that is far from undisputed; whether secondary stress marking is indeed indicated, but only in very subtle ways, is a contentious issue. Thus one must conclude that both instances of "non-$GP^{\lhd}$ phenomena" involve a considerable amount of uncertainty. The apparent scarcity of conclusive evidence for mechanisms in natural language phonology that go beyond $GP^{\lhd}$ is somewhat unexpected given that both SPE and OT are significantly more powerful (see Kaplan and Kay 1994 and Frank and Satta 1998, respectively).

3.4 Further Parameters

3.4.1 Feature Systems

The restriction to privative features is immaterial. A set of PEs is denoted by some propositional formula over $\mathscr{E}$, and the boolean closure of $\mathscr{E}$ is isomorphic to $\wp(\mathscr{E})$. But Keenan (2008:81–109) shows that a binary feature system using a set of features $\mathscr{F}$ can be modeled by the powerset algebra $\wp(\mathscr{F})$, too. So if $|\mathscr{E}| = |\mathscr{F}|$, then $\wp(\mathscr{E}) \cong \wp(\mathscr{F})$, whence the two feature systems are isomorphic. The same result holds for systems using more than two feature values, provided their number

is finitely bounded, since multivalued features can be replaced by a collection of binary valued features given sufficient cooccurrence restrictions on feature values (which can easily be formalized in propositional logic).

One might argue, though, that the core restriction of privative feature systems does not arise from the feature system itself but from the methodological principle that absent features, i.e. negative feature values, behave like constituency information and cannot spread. In general, though, this is not a substantial restriction either, as for every privative feature system $\mathscr{E}$ we can easily design a privative feature system $\mathscr{F} := \{e^+, e^- \mid e \in \mathscr{E}\}$ such that $\mathfrak{M}, w \models e^+$ iff $\mathfrak{M}, w \models e$ and $\mathfrak{M}, w \models e^-$ iff $\mathfrak{M}, w \models \neg e$. Crucially, though, this does not entail that the methodological principle described above has no impact on expressivity when the set of features is fixed across all theories, which is an interesting issue for future research.

3.4.2 Syllable Template

While GP's syllable template could in principle be generalized to arbitrary numbers and sizes of constituents, a look at competing theories such as SPE and Strict CV (Lowenstamm 1996; Scheer 2004) shows that the number of different constituents is already more than sufficient. This is hardly surprising, because GP's syllable template is modeled after the canonical syllable template, which in general is not thought to be in need of further refinement. Consequently, we only need to abandon the connection between constituents and the realization of a segment, lift the restriction on the branching factor and allow theories not to use all three constituent types to obtain a generalized version of the syllable template than can accommodate all three theories. In this generalized version, N does not denote the class of vowels, but the class of segments that need not be licensed but can be licensors. Correspondingly, O designates the class of segments that have to be

licensed if they themselves are licensors, and C the class of segments that always have to be licensed. SPE then operates with a single N constituent of unbounded size, whereas Strict CV uses N and O constituents of size 1.

But now that a constituent may dominate more than two skeletal nodes, government has to be generalized, too. The idea is to let every theory fix the branching factor b for each constituent and the maximum number l of licensees per head. Every node within some constituent has to be licensed by the *head*, i.e. the leftmost node of said constituent. Similarly, all nodes in a coda or non-head position have to be licensed by the head of the following constituent. For every head the total number of licensees may not exceed l.

Even from this informal sketch it should already be clear that the syllable template can have a negative impact on expressivity (canonical GP cannot have, say, a cluster of 7 consonants), but only under the right conditions. For instance, if our feature system is set up in a way such that every symbol of our alphabet is to be represented by a PE in N (as happens to be the case for SPE), restrictions on b and l are without effect. It is an interesting question for future research under which conditions the syllable template has a monotonic effect on generative capacity.

3.5 Evaluation

In summing up, we have seen in this section that there might be an empirical motivation for increasing the power of GP up to the level of SPE by tweaking the spreading mechanism (which in a formal sense corresponds to the introduction of new modal operators). Feature privativity and the syllable template have, if at all, only a restrictive impact on expressivity, GP's true locus of power lies in its spreading mechanism. But it is important not to misconstrue these results. In a

certain sense they make very strong claims, namely that only a tiny portion of all algorithmically definable formal languages are generated by SPE, and even fewer by canonical GP. This goes to show that these theories are indeed very restrictive, but also that there is a systematic relation between them and that we can factor out certain parts of those theories in considering their generative capacity. But at the same time the expressivity results are also very weak, because they fail to reflect an important dimension of linguistic reasoning: the naturalness of an analysis.

For instance, it is an easy formal exercise to restrict SPE's binary feature system with feature cooccurence constraints such that the licit matrices are exactly those corresponding to the PEs derived by GP's privative feature calculus. But in linguistics (and any other sciences, I surmise) nobody would pick the former over the latter, as it seems to miss important generalizations and is cumbersome to work with. This is even more obvious in the case of structural constraints, where it might be possible to account for some phenomenon in $GP^{\triangleleft^+}$, say, by writing out long lists of structural configurations and connecting them by various coding tricks, but if there is a short and simple GP^U formula that gets the same job done, it will be the preferred choice.

Inherent to my examples is the assumption that the naturalness of an account is directly reflected in the *complexity* of the formulas representing it. How exactly this complexity is to be measured is as yet an unsolved problem — not because there are no mathematical ways of doing so, but because those that we know all seem promising. Kornai (2009) argues that Kolmogorov complexity provides the right notion, whereas after several discussions with Ed Stabler I have come to the conclusion that a perspective informed by succinctness and data compression results is more promising (see Grohe and Schweikardt 2005 for some relevant results). However things may eventually turn out, it cannot be emphasized enough that this new direction is not orthogonal to the enterprise undertaken here but rather its

natural extension. The mathematical formalization of phonological theories is the foundation on which the more fine-grained complexity results must rest.

At this point it must be admitted that MTP is currently not in a position to represent every phonological theory in a format that is suitable for those explorations in complexity. Representational theories like GP are naturally captured by our declarative, model-theoretic approach, whereas derivational theories like SPE are usually formalized in automata-theoretic terms as rational relations (Kaplan and Kay 1994; Mohri and Sproat 1996), which resist being recast in logical terms due to their closure properties (note that throughout the thesis I made reference only to the output language of SPE rather than its technical machinery). But not everything is lost. For SPE, one can use a coding trick from two-level phonology (Koskenniemi 1983) and use an unpronounced feature like μ to ensure that all derivationally related strings have the same length. SPE can then be interpreted as language over pairs and hence cast in MSO terms, which was successfully done by Vaillette (2003). Fortunately, OT is very similar to SPE on a formal level (Frank and Satta 1998; Karttunen 1998; Jäger 2002), so Vaillette's technique reigns in the two most influential theories of generative phonology at once. Unfortunately, it is unclear how this method could be extended to weaker grammars (because it follows from Thatcher's theorem that the individual component languages might be of greater complexity than the language over pairs itself).

With respect to future research, then, there is a quantitative as well as a qualitative axis along which MTP can be extended. Along the quantitative axis we further the coverage of MTP to new formalisms, whilst along the qualitative axis we move towards more refined results that characterize classes of phonological theories to greater and greater detail with an emphasis on factors besides expressivity.

Conclusion

In this thesis, I sought to demonstrate that MTP provides us with a general framework in which string-based phonological theories can be matched against each other. To this end, I conducted a case study in which I compared two superficially very dissimilar approaches, SPE and Government Phonology. For my description of Government Phonology, I started with a modal logic which despite its restrictions was still perfectly capable of defining this rather intricate theory with its structured feature calculus, peculiar syllabification and empty categories. In fact, we observed that very simple modal formulas are sufficient for all parts of the theory except spreading, which is slightly surprising considering how little attention has been devoted to spreading in the literature.

I then tried to generalize Government Phonology along several axes, some of which readily lent themselves to conclusive results while others didn't. Most importantly, the power of spreading, by virtue of being an indicator of the necessary power of the description language, has an immediate and monotonic effect on generative capacity. This allowed us to push Government Phonology to the level of SPE. Other parameters play only a minor role with respect to expressivity. As mentioned just a few pages earlier, a lot of work remains to be done, but nevertheless I am confident that the model-theoretic perspective will shed new light on long-standing issues in phonology. I hope that linguists will conceive of it as a welcome addition to their analytic toolbox.

APPENDIX A

Mathematical Preliminaries

Let a, b, c, …, x, y, z be (mathematical) objects. Then the *set* of these objects is denoted by $\{a, b, c, \ldots, x, y, z\}$, and a, b, c, …, x, y, z are its *members*. If A is some set, one may write $x \in A$ or $A \ni x$ to indicate that x is a member of A. The notation $x \notin A$ indicates that x isn't a member of A. The *cardinality* of a set is identical to the number of objects it contains. There is exactly one set of cardinality 0, the *empty set*, denoted by $\emptyset$. Sets can also be defined implicitly; for instance, $\{i \mid 0 < i < 5\} = \{1, 2, 3, 4\}$. This is a good moment to point out the difference between the easily confused symbols $=$ and $:=$. The former denotes the familiar notion of equivalence, whereas the latter should be read as "is defined as" or "stands for". In other words, $:=$ assigns names to objects. We could have referred to $\{i \mid 0 < i < 5\}$ and $\{1, 2, 3, 4\}$ by the names A and B, respectively, which in symbols is expressed by $A := \{i \mid 0 < i < 5\}$ and $B := \{1, 2, 3, 4\}$, and then expressed the equivalence more succinctly by $A = B$.

Returning to sets, we say that A is a subset of B, abbreviated $A \subseteq B$, if every member of A is a member of B. Similarly, A is a superset of B, written $A \supseteq B$, if $B \subseteq A$. If both $A \subseteq B$ and $B \subseteq A$, we say that A and B are identical and write $A = B$. If $A = B$ does not hold, we write $A \neq B$. A subset A of B is *proper* ($\subset$) if $A \neq B$, and similarly for supersets ($\supset$). The relative complement of A and B is $A \setminus B := \{x \mid A \ni x \notin B\}$. Given a set A, $\wp(A)$ is its *powerset*, which contains all subsets of A, and only those. For instance, the powerset of $A := \{1, 2, 3\}$ is

$\wp(A) := \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$.

A *tuple* is a set in which all objects are linearly ordered with respect to each other. Hence, while $\{a,b\} = \{b,a\}$, $\langle b,a\rangle \neq \langle a,b\rangle$. For instance, whether a kissed b or the other way round may matter, whereas there is no difference between "a and b kissed" and "b and a kissed". An n-tuple consists of n objects, and 2-tuples are often called *pairs*. Given two sets A and B, their *cartesian product* is the set $A \times B := \{\langle a,b\rangle \mid a \in A \text{ and } b \in B\}$. Given sets $A_1, A_2, \ldots, A_{n-1}, A_n$, we call $R \subseteq A_1 \times A_2 \times \ldots A_{n-1} \times A_n$ an n-ary *relation*. For instance, the "less than"-relation $<$ relates x to all numbers that are greater than it. Thus $\langle 1,3\rangle \in <$, but $\langle 3,1\rangle \notin <$. Whenever it holds for all tuples $\langle a_1, a_2, \ldots, a_{n-1}, a_n\rangle$ and $\langle b_1, b_2, \ldots, b_{n-1}, b_n\rangle$ of R that $a_1 = b_1$, $a_2 = b_2$, $\ldots$, $a_{n-1} = b_{n-1}$ also implies $a_n = b_n$, we call R a *function* from domain $A_1 \times A_2 \times \ldots \times A_{n-1}$ into co-domain A_n, written $f : A_1 \times A_2 \times \ldots \times A_{n-1} \to A_n$. The familiar bracket notation $f(a_1, a_2, \ldots, a_{n-1}) = a_n$ may also be expressed by $f : a_1, a_2, \ldots, a_{n-1} \mapsto a_n$.

APPENDIX B

Proof of Theorem 3.2

In this appendix, I present a proof of Thm. 3.2 as formulated on page 75, which posits that *nati* cannot be defined in $GP^{\triangleleft^+}$. Recall that the crucial challenge was posed by axiom **N1**:

$$\ulcorner \textit{derived } \text{ɳ} \urcorner \rightarrow (U(\ulcorner \text{r} \urcorner \vee \ulcorner \text{ʂ} \urcorner, \neg \ulcorner \textit{coronal} \urcorner \wedge \neg \ulcorner \text{n} \urcorner))$$

In plain English, this formula enforces that a derived ɳ is preceded by an r or ʂ and no coronal may intervene between the two. The subformula $\neg\ulcorner \text{n} \urcorner$ ensures that the first available target is operated on by *nati*. Abstracting away from extraneous details, we may interpret **N1** as a ban against strings containing at least one substring of the form ad^*cd^*b, where a represents nati-triggers, b nati-targets, c nati-intervenors and d everything else. Or approached from the other direction, **N1** is the logical description of the language $L := \overline{\{a,b,c,d\}^* ad^*cd^*b \{a,b,c,d\}^*}$. My proof establishes that L is not definable in $\text{FO}^2[S,<]$, the two-variable fragment of first-order logic enriched with predicates for the successor relation and the less-than relation. As $\text{FO}^2[S,<]$ is equivalent to RTL, which in turn is equivalent to $GP^{\triangleleft^+}$, this establishes the $GP^{\triangleleft^+}$-undefinability of **N1**. The undefinability of *nati* itself then follows from the fact that **N2** and **N3** are compatible with the assumptions made in the proof, at least if phonology operates on entire sentences rather than single words in isolation (see the discussion in Sec. 3.3.2).

My proof relies on pebble games. As the content of this section is presumably of little interest to a general audience of linguists, I grant myself the luxury of assuming a certain familiarity with Ehrenfeucht-Fraïssé games on the reader's part. Curious readers that lack this background are advised to consult Libkin (2004), which is also where the next two definitions as well as Thm. B.3 were taken from virtually unaltered.

Definition B.1. Let $\mathscr{A}$ and $\mathscr{B}$ be two structures over the signature σ (where σ contains no function symbols), A and B the domains of $\mathscr{A}$ and $\mathscr{B}$, and $\langle a_1, \ldots, a_n \rangle$ and $\langle b_1, \ldots, b_n \rangle$ two tuples in A and B, respectively. Then $\{\langle a_1, b_1 \rangle, \ldots, \langle a_n, b_n \rangle\}$ defines a *partial isomorphism* between $\mathscr{A}$ and $\mathscr{B}$ iff the following conditions hold:

- For every $i, j \leq n$, $a_i = a_j$ iff $b_i = b_j$.
- For every constant symbol c from σ, and every $i \leq n$, $a_i = c^{\mathscr{A}}$ iff $b_i = c^{\mathscr{B}}$.
- For every k-ary relation symbol P from σ and every sequence $\langle i_1, \ldots, i_k \rangle$ of (not necessarily distinct) numbers from the interval $[1, n]$, $\langle a_{i_1}, \ldots, a_{i_k} \rangle \in P^{\mathscr{A}}$ iff $\langle b_{i_1}, \ldots, b_{i_k} \rangle \in P^{\mathscr{B}}$.

Definition B.2. Let $\mathscr{A}$ and $\mathscr{B}$ be as above. A 2-*pebble game* over $\mathscr{A}$ and $\mathscr{B}$ is played by the spoiler and the duplicator as follows. The players have two pairs of pebbles $\langle p^1_{\mathscr{A}}, p^1_{\mathscr{B}} \rangle$ and $\langle p^2_{\mathscr{A}}, p^2_{\mathscr{B}} \rangle$. In each move, the following happens:

- The spoiler chooses a structure, $\mathscr{A}$ or $\mathscr{B}$, and a number $i \in \{1, 2\}$. For the description of the other moves, assume that the spoiler has chosen $\mathscr{A}$ — the other case, when the spoiler chooses $\mathscr{B}$, is completely symmetric.
- The spoiler places the pebble $p^i_{\mathscr{A}}$ on some element of $\mathscr{A}$. If $p^i_{\mathscr{A}}$ was already placed on $\mathscr{A}$, this means that the spoiler either leaves it there or removes it

and places it on some other element of $\mathscr{A}$; if $p^i_{\mathscr{A}}$ was not used, it means that the spoiler picks that pebble and places it on an element of $\mathscr{A}$.

- The duplicator responds by placing $p^i_{\mathscr{B}}$ on some element of $\mathscr{B}$.

We denote the 2-pebble game that continues for n rounds by $\mathrm{PG}_2^n(\mathscr{A}, \mathscr{B})$. After each round of the game, the pebbles placed on $\mathscr{A}$ and $\mathscr{B}$ define a relation $F \subseteq A \times B$ such that if $p^i_{\mathscr{A}}$, $i \in \{1, 2\}$, is placed on $a \in A$ and $p^i_{\mathscr{B}}$ is placed on $b \in B$, then $\langle a, b\rangle \in F$. The duplicator has a winning strategy in $\mathrm{PG}_2^n(\mathscr{A}, \mathscr{B})$ iff he can ensure that after each round $j \leq n$, F is the graph of a partial isomorphism. In this case we write $\mathscr{A} \equiv_{2,n} \mathscr{B}$.

Theorem B.3. *Let $\mathscr{A}$, $\mathscr{B}$ and σ as above. Then $\mathscr{A}$ and $\mathscr{B}$ agree on all sentences of the infinitary two-variable first-order logic of quantifier depth up to n iff $\mathscr{A} \equiv_{2,n} \mathscr{B}$.*

Note that if σ is finite, there are — up to logical equivalence — only finitely many first-order sentences of quantifier depth up to n, so in this case we are in fact only dealing with $\mathrm{FO}^2[S, <]$ sentences. A corollary of Thm. B.3 is that in order to show that a structural constraint c is not $\mathrm{FO}^2[S, <]$-definable, it suffices to exhibit two models $\mathscr{A}$ and $\mathscr{B}$ such that $\mathscr{A} \equiv_{2,n} \mathscr{B}$ but only one of the two obeys c. This is the general strategy that underlies the proof presented here, which exploits four weaknesses of $\mathrm{FO}^2[S, <]$:

1. An FO-formula of quantifier rank n cannot count past n, so for sufficiently long strings the spoiler cannot use the number of as, bs or cs to beat the duplicator.

2. An FO-formula of quantifier rank n cannot take more than n S-steps; metaphorically speaking, its field of vision is bounded. Hence for certain structures the spoiler cannot beat the duplicator by a sequence of S-steps towards a node where $\mathscr{A}$ and $\mathscr{B}$ are not isomorphic.

3. No $<$-move by the spoiler has to be precisely matched by the duplicator — the duplicator may actually jump across more or fewer nodes than the spoiler.

4. Since only two pebble pairs are present at the same time, one of which is replaced per round, the duplicator is very free in his $<$-moves.

The macro-structure of the proof is as follows. Assume that one wants to show that the language L introduced above, which is defined over the alphabet $\Sigma := \{a, b, c, d\}$, is definable in $\mathrm{FO}^2[S, <]$. To simplify the induction step in the proof, we extend $\mathrm{FO}^2[S, <]$ to $\mathrm{FO}^2[S, <, R_\sigma]_{\sigma\in\Sigma}$, which has a dedicated unary relation R_σ for each $\sigma \in \Sigma$ such that node $w \in R_\sigma$ if and only if the label of w is σ. Now for every $\mathrm{FO}^2[S, <, R_\sigma]_{\sigma\in\Sigma}$ formula ϕ of quantifier rank n, one can find strings s and t over the alphabet $\Sigma := \{a, b, c, d\}$ of the form $u_1^\frown a^\frown u_2^\frown c^\frown u_3^\frown b^\frown u_4$ and $u_1^\frown a^\frown u_2^\frown b^\frown u_2^\frown c^\frown u_3^\frown a^\frown u_3^\frown b^\frown u_4$, with $u_1, u_2, u_3, u_4 \in \Sigma^*$, such that $s \models \phi$ iff $t \models \phi$. All u_i, $1 \leq i \leq 4$, are carefully chosen to exploit the weaknesses of $\mathrm{FO}^2[S, <]$ listed above so that the duplicator has a winning strategy. As $s \notin L \ni t$, it follows that L is not $\mathrm{FO}^2[S, <, R_\sigma]_{\sigma\in\Sigma}$-definable, which implies that it is not definable in $\mathrm{FO}^2[S, <]$ either.

The crucial step in the proof is indeed the right choice of s and t. The spoiler has two strategies for winning: He can either use alternating S-moves, where the pebbles remain adjacent in every round (i.e. if p_s^1 is immediately to the left of p_s^2, the spoiler places either p_s^1 immediately to the right of p_s^2 or p_s^2 immediately to the left of p_s^1). As these have to be exactly matched by the duplicator, the spoiler may use them in order to maneuver the pebbles into a position with different labels (as labels are unary relations in $\mathrm{FO}^2[S, <, R_\sigma]_{\sigma\in\Sigma}$, p_s^i and p_t^i must be placed on nodes with identical labels, $i \in \{1, 2\}$). Or he can try to use $<$-moves in such a way that the duplicator has to illegally change direction lest he violate label identity or run out of available targets, e.g. by being too close to the end of the string. The first

strategy can be rendered useless by making the distances between any as, bs and cs sufficiently large. To invalidate the second, one has to ensure that at any position w where s and t are not isomorphic, there are enough as, bs and cs somewhere to the left and somewhere to the right of w.

Let us put this general plan into action now. We construct the $\mathrm{FO}^2[S, <, R_\sigma]_{\sigma\in\Sigma}$-indistinguishable strings s and t from two basic templates as follows. We first define strings u and v over the extended alphabet $\Sigma \cup \{\spadesuit, \clubsuit\}$.

$$u := \overbrace{\underbrace{\dots}_{4\alpha+1} a \underbrace{\dots}_{4\alpha+1} b \underbrace{\dots}_{4\alpha+1} a \underbrace{\dots}_{4\alpha+1} c \underbrace{\dots}_{4\alpha+1} a}^{\alpha} \underbrace{\dots}_{4\alpha+1} b \underbrace{\dots}_{4\alpha+1}$$

$$v := a \underbrace{\dots}_{2\alpha} \spadesuit \underbrace{\dots}_{2\alpha} c \underbrace{\dots}_{2\alpha} \clubsuit \underbrace{\dots}_{2\alpha} b$$

The nodes represented by the dots are assigned labels drawn from $\Sigma \setminus \{a, b, c\}$, so in the case at hand $\underbrace{\dots}_{4\alpha+1}$ stands for $d^{4\alpha+1}$ and $\underbrace{\dots}_{2\alpha}$ for $d^{2\alpha}$, where α is some sufficiently large integer.

Let $v[x, y]$ be the string obtained from v by replacing $\spadesuit$ and $\clubsuit$ by the strings x and y, respectively. I write v_1 and v_2 to denote the substrings x and y of the assembled string $v[x, y]$. Then, for σ and τ drawn from $\Sigma \setminus \{a, b, c\}$ (so $\sigma = \tau = d$ for our purposes), set $u' = u^\frown a \underbrace{\dots}_{4\alpha+1} c \underbrace{\dots}_{4\alpha+1}$ and define

$$s := u^\frown v[\sigma, \tau]^\frown u'$$

$$t := u^\frown v[\sigma \underbrace{\dots}_{2\alpha} b \underbrace{\dots}_{2\alpha} \sigma, \tau \underbrace{\dots}_{2\alpha} a \underbrace{\dots}_{2\alpha} \tau]^\frown u'$$

Note that s is a subsequence of t, or the other way round, t is an extension of s obtained by inserting $\underbrace{\dots}_{2\alpha} b \underbrace{\dots}_{2\alpha} \sigma$ and $\underbrace{\dots}_{2\alpha} a \underbrace{\dots}_{2\alpha} \tau$ right after σ and τ in s, respectively. Now assume that each node w in s is assigned an index i such that

w_0 is the leftmost node and if w_j is the successor of w_i, $0 \leq i$, then $j = i + 1$. The indexing is carried over from s to t such that all nodes w that belong to u, u' or v modulo v_1 and v_2 have identical indices in s and t. For the substrings v_1 and v_2 of t, the indices are obtained as follows: Assume that v_1 and v_2 in s are assigned indices i and j. Then for $1 \leq k \leq \alpha + 1$, the k^{th} nodes of v_1 and v_2 in t are assigned index $i + (k - 1)$ and $j + (k - 1)$, respectively, and for $3\alpha + 3 \leq k \leq 4\alpha + 3$, the indices $i - ((4\alpha + 3) - k)$ and $j - ((4\alpha + 3) - k)$, respectively. For all other choices of k, the index is the special symbol $*$.

A node w' is a *correspondent* of w iff they are not nodes of the same string and have the same index. We denote the set of correspondent(s) of w by $\text{cor}(w)$. This notation is extended to pebbles such that if pebble p is placed on node w, then $\text{cor}(p) = \text{cor}(w)$. We say that the pebbles p_s^i and p_t^i, $i \in \{1, 2\}$, are *in sync* iff $\text{cor}(p_s^i) \subseteq \text{cor}(p_t^i)$ or the other way round. For pebbles p_s^i and p_t^i, $i \in \{1, 2\}$, the p^i*-rectangle* is the pair consisting of the shortest substrings r_s and r_t of s and t, respectively, such that r_s has p_s^i and $w \in \text{cor}(p_t^i)$ as its edge nodes and r_t p_t^i and $w' \in \text{cor}(p_s^i)$. If $\text{cor}(p_t^i)$ is the empty set (i.e. if p_t^i is placed on a node with index $*$), let $\text{cor}(p_t^i) = \text{cor}(e)$, where e is the closest node with an index to the left or the right of p_t^i; whether to pick the first node to the left or to the ride is determined by the shortness of r_s. The edges of the strings r_s and r_t are also called the *corners* of the p^i-rectangle. Given a string r of length l and a node w_i, $0 \leq i \leq l$, $w_i + m$ is the last node of the longest substring of r that starts with w_i and contains at most $m + 1$ nodes. We define $w_i - m$ as the symmetric analog in which w_i marks the end of the string. The *m-neighborhood* of node w is the substring spanning from $w - m$ to $w + m$. For the nodes and α-neighborhoods in and surrounding the extensions of v in s and t, I adopt the naming conventions depicted in Fig. B.1 on page 99.

Theorem B.4. *For each $n \in \mathbb{N}$ there are strings s and t as defined above such that*

$s \equiv_{2,n} t$.

Proof. We have to show that the duplicator has a strategy such that for every $0 \leq k \leq n$, the graph of the relation F defined after k rounds of the n-round 2-pebble game does not fail to be a partial isomorphism. This is trivially true for $k = 0$, as no pebbles have been placed yet. For $k \geq 1$, there are two cases to distinguish. Note that I do not explicitly mention cases where the spoiler uses the S-move strategy as described above, but is taken into account for the induction step.

Case 1 Assume that after $k-1$ rounds F is a partial isomorphism, one of p_s^1 and p_t^1 is to be played by the spoiler, and p_s^2 and p_t^2 are in sync.

Isomorphic s to t If the spoiler plays some node from s that belongs to neither $\overleftrightarrow{L}$ nor $\overleftrightarrow{R}$, the duplicator places p_t^1 on $w \in \mathrm{cor}(p_s^1)$.

Isomorphic t to s In the other direction, if the spoiler places p_t^1 on a node that does not belong to B_l^t, A_r^t, $\overleftarrow{L}$, $\overrightarrow{L}$, $\overleftarrow{R}$ or $\overrightarrow{R}$, the duplicator matches the spoilers move by placing p_s^1 on $w \in \mathrm{cor}(p_t^1)$.

It is evident that after k rounds F still is a partial isomorphism, as only nodes from the isomorphic parts of s and t were played. Moreover, the pebbles are still in sync, so that the same strategy may apply in the next round.

Initial s to t (green) If no pebbles are placed in $\overleftrightarrow{L}$, $\overleftarrow{L}$ or $\overrightarrow{L}$ and the spoiler places p_s^1 on a node in $\overleftrightarrow{L}$, then the duplicator places p_t^1 on $w \in \mathrm{cor}(p_s^1)$ in $\overleftarrow{L}$ or $\overrightarrow{L}$ such that p_t^1 is in $\overleftarrow{L}$ ($\overrightarrow{L}$) if p_s^1 is left of (right of) $a_l^s + (2\alpha + 1)$. If it is neither, the duplicator may choose freely between the correspondents in $\overleftarrow{L}$ and $\overrightarrow{L}$.

Initial s to t (red) Analogous to the previous rule.

Initial t to s (green) If no pebbles are placed in $\overleftrightarrow{L}$, $\overleftarrow{L}$ or $\overrightarrow{L}$ and the spoiler places p_t^1 on a node in $\overleftarrow{L}$ ($\overrightarrow{L}$), then the duplicator puts p_s^1 on $a_l^s + (2\alpha + 1)$.

Initial t to s (red) Analogous to the previous rule.

These strategies clearly preserve partial isomorphism. Note in particular that since the pebbles were in sync in round $k-1$, none of them were placed in B_l^t or A_r^t. Thus they were placed in the isomorphic substrings of s and t, from which it follows immediately that whenever the spoiler is given a choice between two correspondents, both are in fact accessible (because the remaining pebble in t is either to the right or to the left of both correspondents). However, the pebbles are not guaranteed to be in sync anymore. Nevertheless the remaining rules for the game where all pebbles are in $\overleftrightarrow{L}$, $\overleftarrow{L}$ or $\overrightarrow{L}$ ($\overleftrightarrow{R}$, $\overleftarrow{R}$ and $\overrightarrow{R}$) are listed here, as we are dealing with a fairly limited case of being out of sync that does not bring about any wide-ranging complications.

Moving around (green) Suppose p_t^2 is placed in $\overleftarrow{L}$ or $\overrightarrow{L}$. If the spoiler places p_t^1 on a node in $\overleftarrow{L}$ or $\overrightarrow{L}$ to the left (the right) of p_t^2, then, if p_s^2 is placed in $\overleftrightarrow{L}$, the duplicator places p_s^1 on the node closest to p_s^2 such that partial isomorphism is preserved; otherwise, he places p_s^1 on $w \in \mathrm{cor}(p_t^2)$ such that partial isomorphism is preserved. The same strategy holds in the other direction.

Moving around (red) Analogous to the previous rule.

For the second configuration, preservation of partial isomorphism depends on two assumptions (only given for one of the directions here):

- p_s^2 is located in a position such that the placement of p_s^1 does not violate partial isomorphism, in particular the directionality of the spoiler's move.

- p_s^2 has the same label as p_t^2.

The latter follows immediately from our induction hypothesis, whereas the former follows from our strategy so far. As for the first configuration, we only have to show that the intended closest node exists. In the simplest case, where the spoiler commits to an S-move, this is obvious. In the case of an $<$-move, it is the node two S-moves away from p_s^2 (in the relevant direction). A special case warrants explicit mention: if the spoiler positions p_t^1 to the left (the right) of p_t^2 for several rounds without interruption, the duplicator puts p_s^1 on the same node in each round. Note that if the target of the duplicator's move is not in $\overleftrightarrow{L}$, $\overleftarrow{L}$ or $\overrightarrow{L}$, then it must be in A_l^s, C^s, A_l^t, B_l^t or C^t. But as the duplicator did a minimal move and α was chosen to be a sufficiently large integer, the target cannot be a_l^s, c^s, a_l^t b_l^t or c^t, whence it is labeled d and partial isomorphism is still preserved. The reasoning given applies to the second rule without notable modifications. For the sake of convenience we define a strategy for the special case where the spoiler forces the duplicator into B_l^t or A_r^t. All other continuations are covered by Case 2.

Avoid B_l^t If *Moving around (green)* would lead the duplicator to place p_t^1 on node w in B_l^t and p_s^1 is to the right (the left) of p_s^2, then he puts it on $w+(2\alpha+1)$ ($w-(2\alpha+1)$) instead.

Given the layout of t, $w+(2\alpha+1)$ is guaranteed to have the same label as w, so partial isomorphism is preserved.

For the last subcase, assume that in round k the spoiler places p_t^1 on the node $b_l^t - x$, $0 \leq x \leq \alpha$. Then the duplicator proceeds as follows:

Initial B_l^t If p_s^2 is already placed on $b_1^s - x$ or, alternatively, on a node to its right while p_t^2 is left of p_t^1, the duplicator places p_s^1 on $b_2^s - x$. Else, the duplicator places p_s^1 on $b_1^s - x$.

The strategies for $b_l^t + x$ as well as a_r^t are symmetric and will be skipped here. We observe that since only one position in each string can be occupied by a pebble whenever the other one is to be placed, at least one of the two, $b_1^s - x$ or $b_2^s - x$, is always available. Also, given the duplicators strategy so far, if p_t^2 is to the left (the right) of B_l^t, p_s^2 is guaranteed to be to the left of B_2^s (the right of B_1^s), whence p_s^1 can be placed on a node belonging to one of the two without breaking the partial isomorphism. But no matter which node the duplicator chooses, p_s^1 and p_t^1 are no longer in sync, which brings us to the second case.

Case 2 Assume that after $k-1$ rounds F is a partial isomorphism, one of p_s^1 and p_t^1 is to be played by the spoiler, and p_s^2 and p_t^2 are not in sync.

Outside rectangle If the spoiler places p_s^1 (p_t^1) outside the p^2-rectangle, the duplicator uses one of the strategies listed under Case 1.

If none of the corners of the p^2-rectangle lie outside the domain of *Isomorphic s to t* or *Isomorphic t to s*, then all previously introduced strategies are applicable without further modifications and are guaranteed to preserve partial isomorphism. Otherwise, one of the corners must lie in $\overleftrightarrow{L}$ or $\overleftrightarrow{R}$. In this case, both *Isomorphic s to t* and *Isomorphic t to s* will preserve partial isomorphism if they are applicable. If this strategy isn't applicable, then the spoiler has placed a pebble on some node in $\overleftrightarrow{L}$, $\overleftarrow{L}$, $\overrightarrow{L}$, $\overleftrightarrow{R}$, $\overleftarrow{R}$ or $\overrightarrow{R}$. Assume that the pebble was placed on some node belonging to $\overleftrightarrow{L}$, $\overleftarrow{L}$ or $\overrightarrow{L}$. Then either no node from these substrings is contained in the rectangle, or at least one is not. The former case is fully covered by the strategies given for Case 1. So consider the latter case. Assume w is the node not contained in the rectangle. Then it follows from the definition of a p^i-rectangle that some $w' \in \mathrm{cor}(w)$ is not contained in the rectangle either. Hence *Moving around (green)* can be applied, and from the definition of p^i-rectangles it follows that p_s^2

and p_t^2 are either to the left or to the right of p_s^1 and p_t^1, respectively. Hence partial isomorphism is preserved.

Inside rectangle Suppose the spoiler places p_s^1 inside the p^2-rectangle to the left of p_s^2 (hence $w \in \mathrm{cor}(p_t^2)$ is left of p_s^2, too). Then the duplicator puts p_t^1 on a node to the left of p_t^2 such that for $\sigma \in \{a, b, c\}$

1. if p_s^1 is placed on $\sigma + x$ or $\sigma - x$, $x \leq \alpha$, then p_t^1 is placed on $\sigma + x$ or $\sigma - x$, respectively, for the closest σ to the left of p_t^2.
2. else, p_t^1 is put on the closest d-labeled node to the left of p_t^2 that is at least $\alpha + 1$ steps away from any σ-labeled node.

The strategy with reversed directions and the strategies for when spoiler picks p_t^1 are analogous.

In case (1), the spoiler places a pebble so close to an a, b or c that he could potentially reach it with S-moves before the game is over. The duplicator reacts by putting a pebble at the same distance from the closest a, b or c, mirroring the spoiler's move and thus preserving isomorphism. Crucially, such a symbol is guaranteed to exist: Pebbles may get out of sync only in $B_l^t - \beta_k$ and $A_r^t - \beta_k$ or at the edges of $\overleftrightarrow{L}$, $\overleftarrow{L}$, $\overrightarrow{L}$, $\overleftrightarrow{R}$, $\overleftarrow{R}$, $\overrightarrow{R}$ (as usual, I discuss only one of the configurations as they are all very similar). Considering the duplicator's strategy developed in Case 1, there are two worst-case scenarios: First, the spoiler may play $b_l^t \pm x$ in the first round (where $0 \leq x \leq \alpha$), which the duplicator matches by playing $b_1^s \pm x$. Then the only interesting move for the spoiler is to play a node from A_l^s, A_l^t or B_l^t. By taking the shortest possible $<$-moves from a to a (alternatively, from b to b), the spoiler may hope to reach the end of the string so that the duplicator is faced with a losing position in which the spoiler pebbles an a or a b in one string and there are no such symbols left in the other. However, the spoiler must not skip a

possible target site in his $<$-moves, otherwise he will jump so far that the duplicator can sync the pebbles. In n rounds, then, he can only move n symbols of one kind in either direction. For each kind, though, there are at least n symbols to the left as well as to the right of both B_l^t and B_1^s, so this strategy cannot come to fruition before the game is over. In the second worst-case scenario, the spoiler first blocks $b_1^s \pm x$ by a pebble before putting another on $b_l^t \pm x$ in the next round, which is followed by the duplicator moving a pebble onto $b_2^s \pm x$. Now the distance with respect to b-symbols is bigger between the pebbles, but at the same time there are only $n-2$ rounds left to play, whence there are still enough symbols left at either side. In sum, then, (1) also preserves the partial isomorphism and does not open up an opportunity for the spoiler to checkmate the duplicator in n rounds or less.

Substrategy (2) just tells the duplicator that if the spoiler picks a node labeled by d that is sufficiently far away from all as, bs and cs so that they cannot be reached by S-moves in the remaining rounds of the game, then take the shortest $<$-move to a d-labeled node that is similarly far away from the other symbols. Evidently this move also preserves partial isomorphism (and emphasizes that placing a pebble on such d-nodes is always a waste for the spoiler). □

In closing, I want to point out that the proof generalizes to any alphabet $\Sigma \subseteq \{a, b, c, d\}$, as s and t would still be strings over this alphabet. However, the reliance on long spans of identical symbols seems out of place for a proof on natural language phonology, where constraints are conjectured to exist that proactively penalize too much repetition. This is commonly referred to as OCP effects in the literature. However, the assumption of the proof is less about actual symbol distributions in the string but rather that whatever other symbols the alphabet may contain, their distribution has no implications for the distribution of a, b and c. This seems plausible for *nati*, where the only conditions are those mentioned in the

constraints. Readers that are unhappy with this situation, though, are invited to come up with a proof for richer alphabets. All that is necessary is to construct the strings such that whatever clues about the current position in a string one might be able to derive from some specific symbol is rendered useless by the form of the string itself. In practical terms, this would mean that every neighborhood and every string between the neighborhoods has to appear sufficiently often in all possible combinations so that the duplicator can navigate around the strings unharmed for at least n rounds.

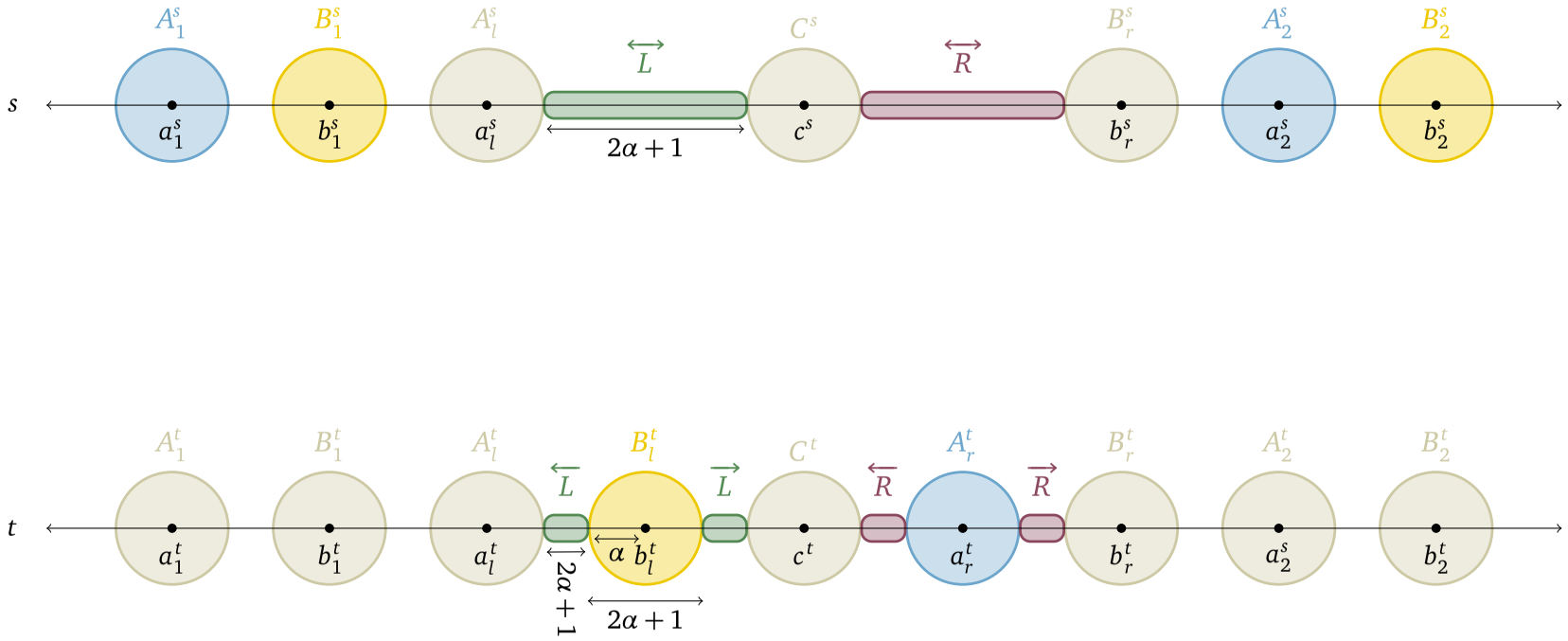


Figure B.1: Naming conventions used for parts of the strings s and t

BIBLIOGRAPHY

Albro, Dan. 2005. *Studies in computational optimality theory, with special reference to the phonological system of Malagasy*. Doctoral Dissertation, University of California, Los Angeles.

Blackburn, Patrick, Maarten de Rijke, and Yde Venema. 2002. *Modal logic*. Cambridge: Cambridge University Press.

Brody, Michael. 1995. *Lexico-logical form: A radically Minimalist theory*. Cambridge, Mass.: MIT Press.

Brody, Michael. 2002. On the status of representations and derivations. In *Derivation and explanation in the minimalist program*, ed. Samuel D. Epstein and Daniel T. Seely, 19–41. Oxford: Blackwell.

Browning, Marguerite. 1989. ECP $\neq$ CED. *Linguistic Inquiry* 20:481–491.

Büchi, J. Richard. 1960. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 6:66–92.

Charette, Monik. 1990. Licence to govern. *Phonology* 7:233–253.

Charette, Monik, and Asli Göksel. 1994. Switching and vowel harmony in Turkic languages. *SOAS Working Papers In Linguistics and Phonetics* 4:31–52.

Charette, Monik, and Asli Göksel. 1996. Licensing constraints and vowel harmony in Turkic languages. *SOAS Working Papers In Linguistics and Phonetics* 6:1–25.

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

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 1981. *Lectures on government and binding: The Pisa lectures*. Dordrecht: Foris.

Chomsky, Noam. 1986. *Barriers*. Cambridge, Mass.: MIT Press.

Chomsky, Noam, and Morris Halle. 1968. *The sound pattern of English*. New York: Evanston.

Cohen, Joelle. 1991. On the expressive power of temporal logic for infinite words. *Theoretical Computer Science* 83:301–312.

Cohen, Joelle, Dominique Perrin, and Jean-Eric Pin. 1993. On the expressive power of temporal logic. *Journal of Computer and System Sciences* 46:271–294.

Etessami, Kousha, Moshe Y. Vardi, and Thomas Wilke. 1997. First-order logic with two variables and unary temporal logic. In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science*, 228–235.

Frank, Robert, and Giorgio Satta. 1998. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24:307–315.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, and Ivan A. Sag. 1985. *Generalized phrase structure grammar*. Oxford: Blackwell.

Goldsmith, John. 1976. *Autosegmental phonology*. Doctoral Dissertation, MIT.

Graf, Thomas. 2010a. Comparing incomparable frameworks: A model theoretic approach to phonology. *University of Pennsylvania Working Papers in Linguistics* 16:Article 10. Available at: http://repository.upenn.edu/pwpl/vol16/iss1/10.

Graf, Thomas. 2010b. Formal parameters of phonology: From government phonology to SPE. In *Interfaces: Explorations in logic, language and computation*, ed.

Thomas Icard and Reinhard Muskens, volume 6211 of *Lectures Notes in Artifical Intelligence*, 72–86. Berlin: Springer.

Grohe, Martin, and Nicole Schweikardt. 2005. The succinctness of first-order logic on linear orders. *Logical Methods in Computer Science* 1:Paper 6.

Haas, Mary R. 1977. Tonal accent in Creek. In *Southern California occasional papers in linguistics*, ed. Larry M. Hyman, volume 4, 195–208. Los Angeles: University of Southern California. Reprinted in Sturtevant (1987).

Halpern, Joseph Y. 1995. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *Artificial Intelligence* 75:361–372.

Hansson, Gunnar Ólafur. 2001. *Theoretical and typological issues in consonant harmony*. Doctoral Dissertation, University of California, Berkeley.

Harris, John, and Geoff Lindsey. 1995. The elements of phonological representation. In *Frontiers of phonology*, ed. Jacques Durand and Francis Katamba, 34–79. Harlow, Essex: Longman.

Hayes, Bruce. 1995. *Metrical stress theory*. Chicago: Chicago University Press.

Immerman, Neil, and Dexter C. Kozen. 1989. Definability with bounded number of bound variables. *Information and Computation* 83:121–139.

Infante-Lopez, Gabriel G., Carlos Areces, and Maarten de Rijke. 2003. Controlled model exploration. *Advances in Modal Logic* 4:1–16.

Jensen, Sean. 1994. Is ʔ an element? Towards a non-segmental phonology. *SOAS Working Papers In Linguistics and Phonetics* 4:71–78.

Johnson, C. Douglas. 1972. *Formal aspects of phonological description*. The Hague: Mouton.

Johnson, David, and Shalom Lappin. 1997. A critique of the minimalist program. *Linguistics and Philosophy* 20:273–333.

Jäger, Gerhard. 2002. Gradient constraints in finite state OT: The unidirectional and the bidirectional case. In *More than words. A festschrift for Dieter Wunderlich*, ed. I. Kaufmann and B. Stiebels, 299–325. Berlin: Akademie Verlag.

Kaplan, Ronald M., and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378.

Karttunen, Lauri. 1998. The proper treatment of optimality in computational phonology. Manuscript, Xerox Research Center Europe.

Kaye, Jonathan. 1992. Do you believe in magic? The story of s+C sequences. *Working Papers in Linguistics and Phonetics* 2:293–313.

Kaye, Jonathan. 1995. Derivations and interfaces. In *Frontiers of phonology*, ed. Jacques Durand and Francis Katamba, 289–332. London: Longman.

Kaye, Jonathan. 2000. A user's guide to government phonology. URL http://134.59.31.7/~scheer/scan/Kaye00guideGP.pdf, unpublished manuscript.

Kaye, Jonathan. 2001. Working with licensing constraints. Ms., Guangdong University of Foreign Studies.

Kaye, Jonathan, Jean Lowenstamm, and Jean-Roger Vergnaud. 1985. The internal structure of phonological representations: a theory of charm and government. *Phonology Yearbook* 2:305–328.

Kaye, Jonathan, Jean Lowenstamm, and Jean-Roger Vergnaud. 1990. Constituent structure and government in phonology. *Phonology Yearbook* 7:193–231.

Keenan, Edward. 2008. Mathematical structures in language. Ms., University of California, Los Angeles.

Kisseberth, Charles. 1970. On the functional unity of phonological rules. *Linguistc Inquiry* 1:291–306.

Kobayashi, Masato. 2004. *Historical phonology of old Indo-Aryan consonants*. Tokyo: ILCAA.

Kobele, Gregory M. 2006. *Generating copies: An investigation into structural identity in language and grammar*. Doctoral Dissertation, UCLA.

Kornai, Andras. 2009. The complexity of phonology. *Linguistic Inquiry* 40:701–711.

Kornai, Andras, and Geoffrey K. Pullum. 1990. The X-bar theory of phrase structure. *Language* 66:24–50.

Koskenniemi, Kimmo. 1983. Two-level morphology: A general computational model for word-form recognition and production. Publication 11, University of Helsinki, Department of General Linguistics, Helsinki.

Kracht, Marcus. 2003. Features in phonological theory. In *Foundations of the formal sciences II, applications of mathematical logic in philosophy and linguistics*, ed. Benedikt Löwe, Wolfgang Malzkorn, and Thomas Räsch, number 17 in Trends in Logic, 123–149. Dordrecht: Kluwer. Papers of a conference held in Bonn, November 11–13, 2000.

Krohn, Kenneth, and John Rhodes. 1965. Algebraic theory of machines. I. Prime

decomposition theorem for finite semigroups. *Transactions of the American Mathematical Society* 116:450–464.

Kröger, Fred, and Stephan Merz. 2008. *Temporal logic and state systems*. Berlin: Springer.

Leben, William. 1973. *Suprasegmental phonology*. Doctoral Dissertation, MIT.

Leucker, Martin, and César Sánchez. 2005. Regular linear temporal logic. In *Proceedings of The 4th International Colloquium on Theoretical Aspects of Computing (ICTAC'07)*, number 4711 in Lecture Notes in Computer Science, 291–305.

Libkin, Leonid. 2004. *Elements of finite model theory*. Berlin: Springer.

Lowenstamm, Jean. 1996. CV as the only syllable type. In *Current Trends in Phonology: Models and Methods*, ed. Jacques Durand and Bernard Laks, 419–421. European Studies Research Institute, University of Salford.

Macdonell, Arthur. 1910. *Vedic grammar*. Strassburg: Trübner.

McCarthy, John J. 1979. On stress and syllabification. *Linguistic Inquiry* 10:443–465.

McNaughton, Robert, and Seymour Pappert. 1971. *Counter-free automata*. Cambridge, Mass.: MIT Press.

Meyer, Albert R. 1975. Weak monadic second-order theory of successor is not elementary recursive. In *Logic colloquium: Symposium on logic 1972–1973*, ed. R. Parikh, volume 453 of *Lecture Notes in Mathematics*, 132–154. Berlin: Springer.

Michaelis, Jens. 2001. Transforming linear context-free rewriting systems into minimalist grammars. *Lecture Notes in Artificial Intelligence* 2099:228–244.

Mitchell, Terence F. 1960. Prominence and syllabification in Arabic. *Bulletin of the School of Oriental and African Studies* 23:369–389.

Mohri, Mehryar, and Richard Sproat. 1996. An efficient compiler for weighted rewrite rules. In *In 34th Annual Meeting of the Association for Computational Linguistics*, 231–238.

Pöchtrager, Markus A. 2006. *The structure of length*. Doctoral Dissertation, University of Vienna.

Potts, Christopher, and Geoffrey K. Pullum. 2002. Model theory and the content of OT constraints. *Phonology* 19:361–393.

Pullum, Geoffrey K. 2007. The evolution of model-theoretic frameworks in linguistics. In *Model-Theoretic Syntax @ 10*, ed. James Rogers and Stephan Kepser, 1–10.

Pullum, Geoffrey K., and James Rogers. 2006. Animal pattern-learning experiments: Some mathematical background. Ms., Radcliffe Institute for Advanced Study, Harvard University.

Rizzi, Luigi. 1990. *Relativized minimality*. Cambridge, Mass.: MIT Press.

Rogers, James. 1996. A model-theoretic framework for theories of syntax. In *Proceedings of the 34th Annual Meeting of the ACL*, 10–16. Santa Cruz, USA.

Rogers, James. 1998. *A descriptive approach to language-theoretic complexity*. Stanford: CSLI.

Scheer, Tobias. 2004. *A lateral theory of phonology: What is CVCV and why should it be?*. Berlin: Mouton de Gruyter.

Schein, Barry, and Donca Steriade. 1986. On geminates. *Linguitic Inquiry* 17:691–744.

Sommerstein, Alan. 1974. On phonotactically motivated rules. *Journal of Linguistics* 10:71–94.

Stabler, Edward P. 1992. *The logical approach to syntax: Foundations, specifications and implementations of theories of government and binding*. Cambridge, Mass.: MIT Press.

Sternefeld, Wolfgang. 1996. Comparing reference-sets. In *The role of economy principles in linguistic theory*, ed. Chris Wilder, Hans-Martin Gärtner, and Manfred Bierwisch, 81–114. Berlin: Akademie Verlag.

Sturtevant, William C., ed. 1987. *A Creek source book*. New York: Garland.

Thatcher, James W. 1967. Characterizing derivation trees for context-free grammars through a generalization of finite automata theory. *Journal of Computer and System Sciences* 1:317–322.

Thomas, Wolfgang. 1979. Star-free regular sets of ω-sequences. *Information and Control* 42:148–156.

Vaillette, Nathan. 2003. Logical specification of regular relations for NLP. *Natural Language Engineering* 9:65–85.

Vardi, Moshe Y. 1988. A temporal fixpoint calculus. In *Proceedings of the 15th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 250–259.

Weil, Pascal. 2004. Algebraic recognizability of languages. In *Mathematical foundations of computer science 2004*, ed. Jiri Fiala, Václav Koubek, and Jan Kratochvíl, volume 3153 of *Lecture Notes in Computer Science*, 149–175. Berlin: Springer.

Whitney, William D. 1889. *Sanskrit grammar*. London: Oxford University Press.