

Comparing Incomparable Frameworks

A Model Theoretic Approach to Phonology

1 Model Theoretic Approach

1.1 The Basic Idea

Linguistic theory = set of logical formulas

Grammatical structure = *model* of set of logical formulas

A structure is licensed (a *model*) iff it satisfies all formulas in that set (see Figure 1 and 2).

Linguistic Constraint	Logical Formula
No segment associated with a low tone has a high tone	$L \rightarrow \neg H$ or equivalently $\forall x[L(x) \rightarrow \neg H(x)]$
<i>Rootedness</i> : Every tree has a unique root	$\exists x \forall y[\textit{dominates}(x, y)]$

Table 1: Two linguistic constraints and their respective logical formulas

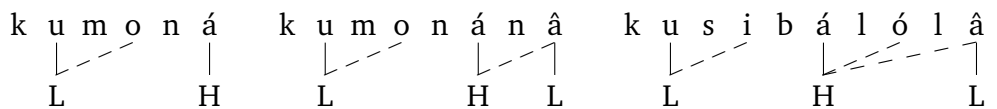


Figure 1: $\neg H \rightarrow L$ is satisfied by all structures, but $L \rightarrow \neg H$ only by the leftmost one

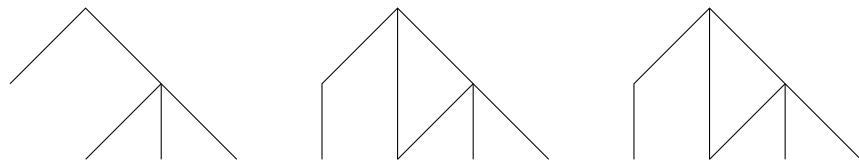


Figure 2: *Rootedness* is satisfied by all structures but the rightmost one

Why would we do this? Because certain properties of linguistic theories correlate to the properties of the logic in which it is formalized.

1.2 Previous Applications in Syntax

Both foundational work on feature matrices (Blackburn 1993; Blackburn and Spaan 1993; Blackburn 1994) and tree structures (Blackburn et al. 1993; Blackburn and Meyer-Viol 1994;

Backofen et al. 1995; Rogers 2003) and applied work on GB (Stabler 1992; Rogers 1998a), GPSG (Rogers 1997a) and TAG (Rogers 1997b, 1998b,c); Potts and Pullum (2002) apply the framework to phonology and study the expressivity of various constraint types in OT

- (1) *Two interesting results of Model Theoretic Syntax*
 - a. Independence of GB modules (Stabler 1992)
CED effects of the barriers theory depend on the ECP, but not on Subjacency
Linguistic discussion: Chomsky (1986), Browning (1989)
 - b. Universals in GB and GPSG (Rogers 1997a)
Universals in GB are restrictions on tree structures (e.g. \bar{X} -theory, ECP), but universals in GPSG are closure properties of sets of tree structures (i.e. conditions on entire languages)
Linguistic discussion: previously unnoticed!

1.3 Linguistic Theory versus Model Theory

- (2) *Advantages of the linguistic approach*
 - a. **Proven and tested**
 - b. **Immediate linguistic relevance**
 - c. **Requires only linguistic skills**
- (3) *Disadvantages of the linguistic approach*
 - a. **Labor intensive**
 - b. **High risk of misanalysis**
There might be a non-obvious way to account for the data which we forgot to consider, endangering our results.
 - c. **Dependent on empirical data**
Because of the potential insufficiency of data, the equivalence of theories cannot be established conclusively.
 - d. **Implementation specific**
Our results probably won't carry over to modifications or even syntactic variants of the studied theories.
- (4) *Advantages of the model theoretic approach*
 - a. **Rigorous expressivity results**
If a linguistic theory T can be formalized in a logic L , whatever cannot be defined in L cannot be defined in T .
 - b. **Classification system**
All theories formalizable in a logic L share certain properties and hence form a class that can be studied independently, saving us a lot of work.
 - c. **Abstraction**
By focusing on structures rather than the technical machinery generating them, we can blend out details that obscure essential linguistic claims (cf. Rogers's result on universals in GB and GPSG).
 - d. **Modularization**
Since theories are sets of formulas, we can restrict our attention to specific formulas representing some linguistic module and study them in isolation, or add them to another theory and calculate the results.

- e. **Consistency**
We can check linguistic theories for consistency and redundancies (cf. Stabler's independence results for GB).
- f. **Bridge to psycholinguistics**
Mathematical logic is closely related to automata theory and complexity theory, giving us hints concerning the runtime-behavior and memory requirements of a theory.

2 A Case Study: Government Phonology, SPE, and Their Expressive Power

2.1 Why Government Phonology, and what is it?

Government Phonology (GP) (Kaye et al. 1985, 1990) aims to be a maximally restricted theory of phonology. It deviates from SPE in various interesting ways, making it difficult to compare the two:

- (5) *Overview of basic properties of GP*
 - a. **Representational** (SPE: derivational)
just like in OT, there are no sequentially ordered structure changing operations but only licensed or unlicensed structures
 - b. **Privative feature system** (SPE: binary)
features have no values, an element either has them or not (like H and L for tone); see Table 2 on the following page
 - c. **Constituent structure** (SPE: strings)
built from onset-rhyme templates with different positions for consonants and vowels; see Figure 4 on page 5
 - d. **Unrealized segments** (SPE: no covert material)
some positions in the phonological string may remain unpronounced if they satisfy certain licensing conditions
 - e. **Autosegmental** (SPE: segmental)
features may spread from one segment to another one; see Figure 3 on the following page

Figure 3 on the next page gives two examples for typical GP structures. The representational perspective poses no challenge to the model theoretic approach. We now show how the other characteristics can be formalized.

2.2 Formalization of Government Phonology

2.2.1 Feature System

The feature calculus can be easily *captured with simple propositional logic* (we will ignore the head-operator distinction here for the sake of brevity). This simple theory is called GP_0^0 , where the subscript tells us how many steps we can move to the left or the right and the superscript how far we can “see” to the left and the right. As we are currently restricted to single segments, both values are 0.

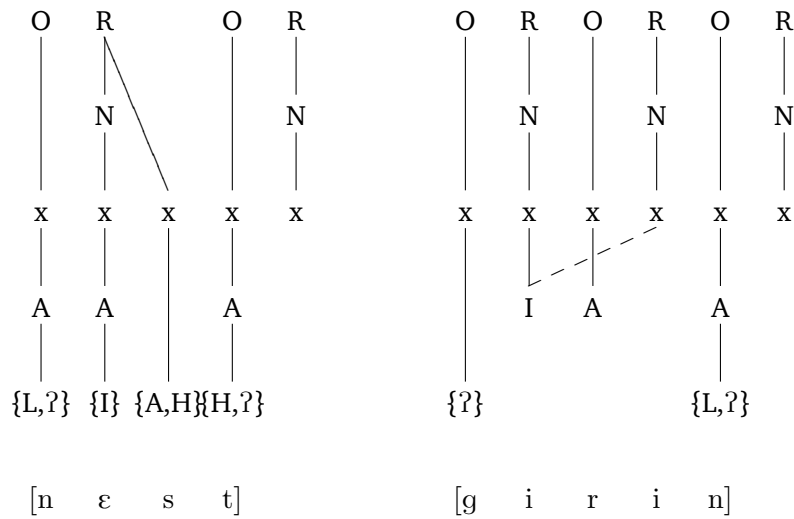


Figure 3: Two examples of annotated phonological structures in GP

r	({ }, A)	a	({ }, A)
j	({ }, I)	i	({ }, I)
w	({ }, U)	u	({ }, U)
g	({?}, _)	ə	({ }, _)
s	({A, H}, _)	e	({A}, I)
n	({L, ?}, A)	ε	({I}, A)

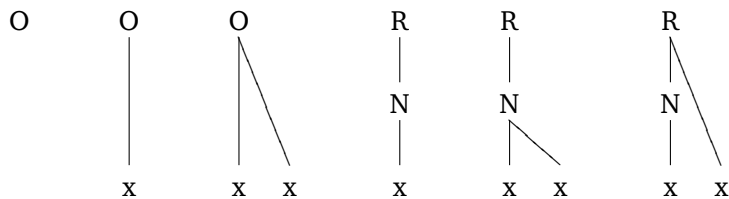
Table 2: Some common phonological expressions

(6) *Propositional logic* p := the segment has feature p $\neg p$:= the segment does not have feature p $p \wedge q$:= the segment has feature p and feature q $p \vee q$:= the segment has feature p or feature q , or both $p \rightarrow q$:= if the segment has feature p , then it also has feature q $p \leftrightarrow q$:= the segment has feature p iff it has feature q

r	$A \wedge \neg I \wedge \neg U \wedge \neg L$
\emptyset	$\neg A \wedge \neg I \wedge \neg U \wedge \neg L$
$\{i, \iota, j, e, \varepsilon\}$	$I \wedge \neg U \wedge \neg L$
$\{a, \alpha, r, \imath, \iota, j, e, \varepsilon\}$	$(A \vee I) \wedge \neg U \wedge \neg L$

Table 3: Logical formulas for defining phonological expressions (A, I, U, L the only features)**2.2.2 Constituent Structure**

These formulas hold at a single segment, but we want to speak about the entire string, of course. In GP, the phonological structure is enriched with a limited kind of syllabic constituency. In total, there are six basic building blocks whose combination is restricted by additional well-formedness conditions.

**Figure 4: The six basic building blocks of phonological structure**(7) *How to combine the building blocks*

- a. Every phonological expression 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.

We first reencode the structures slightly. This doesn't change anything about the explanatory value of our theory or the ontological claims it makes.

- (8) a. Treat rhyme as a mere notational device.
- b. Add an explicit coda position C .
- c. Add a feature f_{ake} for unassociated onsets.
- d. Turn binary branching N/O into two unary branching N/O .

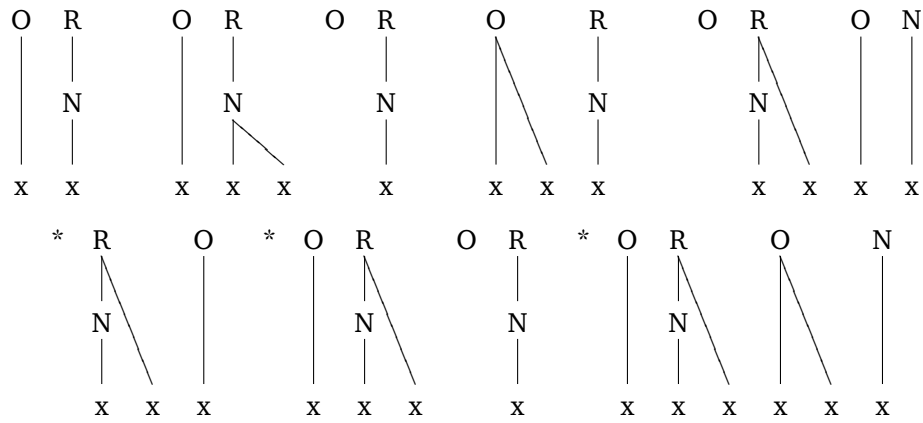


Figure 5: Examples of licit and illicit structures

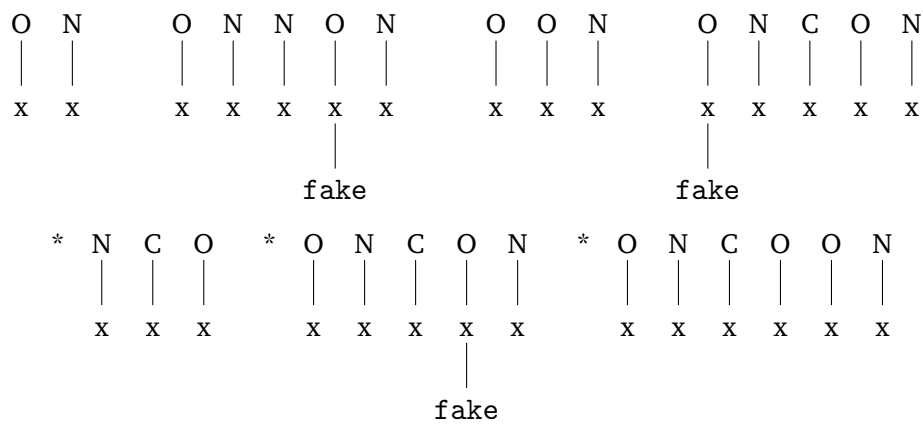


Figure 6: Corresponding examples of licit and illicit structures in simplified notation

To move us along the phonological string, we use two operators \triangleleft and \triangleright that move us one step to the left and the right, respectively (with \triangleleft and \triangleright as their duals). The operators can be nested to move us several steps at once. This gives us GP_n^n , n finite.

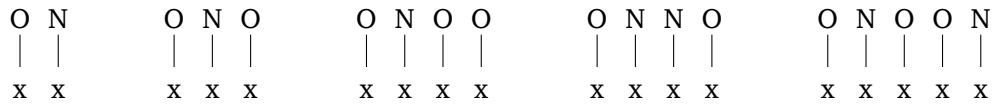


Figure 7: All structures save the two rightmost ones satisfy the formula $(N \rightarrow \triangleleft O) \wedge (\triangleleft \triangleleft O \rightarrow O)$

2.2.3 Unrealized Segments

GP makes crucial use of empty categories, i.e. the assumption that not all constituents in the phonological string have to contain phonological material.

(9) *The phonological ECP*

A p-licensed empty category receives no phonetic interpretation.

(10) *p-licensing*

a. Final Empty Nuclei Parameter (FEN)

Domain-final empty categories are/aren't p-licensed.

b. Magic Licensing

s+consonant sequences license a preceding empty nucleus.

c. Proper Government

Properly governed (empty) nuclei are p-licensed.

(11) *Proper Government*

a properly governs b if

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

b. a is not itself licensed, and

c. neither a nor b are government licensers.

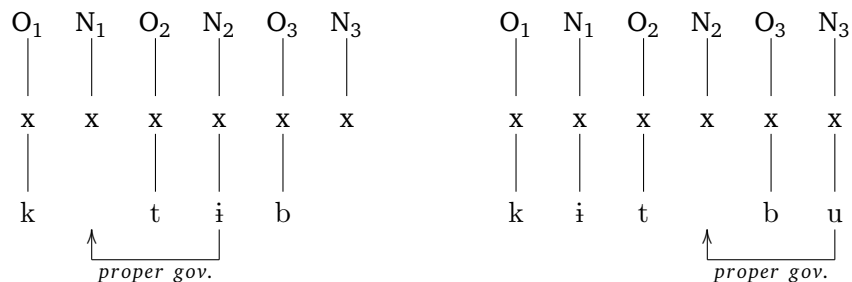


Figure 8: Proper government in Hebrew paradigms

FEN und Magic Licensing are easy to implement. For Proper Government, we observe that in our encoding without branching constituents, this condition is *equivalent to a restriction on*

the number of nodes that may separate the licensed nucleus from its licensing nucleus and the next nucleus (or word boundary) to its right.

(12) Formalization of *p*-licensing

$$N \wedge \checkmark \leftrightarrow \underbrace{\triangleright (C \wedge \bigvee_{i \in S} i)}_{\text{Magic Licensing}} \vee \underbrace{(\neg \triangleleft N \wedge \blacktriangleright \perp)}_{\text{FEN}} \vee \underbrace{((\neg \triangleleft N \rightarrow \triangleleft (\triangleleft N \vee \blacktriangleleft \perp)) \wedge (\neg \triangleright N \rightarrow \triangleright \triangleright (N \wedge \neg \mu)))}_{\text{Proper Government}}$$

2.2.4 Spreading

Elements can spread from one position to another one, e.g. in vowel-harmony. Unfortunately, though, most of the GP literature focuses on the number of features and constituency. As a consequence, the specifics of spreading have never been explicitly defined, so we don't know

- whether spreading is always *obligatory*
- whether its *directionality* is restricted (zig-zag-spreading?)
- what qualifies as a *source* or a *target* for spreading
- how far an element might spread

All possible answers to the first three questions can be accommodated in GP_n^n , but the solution is a little involved. See the appendix for details.

2.3 Solving Theory Internal Problems — The Power of Licensing Constraints

An open question in the GP literature concerns the power of *Licensing Constraints* (LCs), i.e. constraints on the combinations of elements that are licensed. Evidently, we do not want LCs to increase the power of our theory, so we want them to use the weakest logic possible. If we restrict ourselves to propositional logic, we can only restrict combinations of elements without any reference to structural properties (we can, for instance, block *U* from spreading into an expression headed by *I*, but we cannot restrict this blocking to cases where *U* is spread from an expression containing an *A*; we also cannot block *I* from combining with *U* if no adjacent phonological expression contains *I* or *U*). Almost all LCs in the literature can be formalized in propositional logic. The only exception are LCs for tone elements. These invoke a notion of tone domain, which is left undefined. As long as the domain is bounded, such LCs can be captured in the modal logic we have used so far. If its size is unbounded, however, a more powerful logic will be needed. In any case the presence of tonal LCs implies that spreading and other relations could be restricted in very elaborate ways, too.

2.4 Extensions of GP

We add another pair of operators \frown and \smile that allow us to see the entire domain to the left or to the right of the current segment. However, they do not allow us to move there in a principled way (they would carry us to some possible target in the domain, but we can never know at which target we actually arrived). This gives us GP_n^ω , n finite and ω the symbol for infinity.

Adding *U* (until) and *S* (since), we can move freely around the entire string, giving us GP_ω^ω . However, our targeting is still slightly off mark, as we cannot, for instance, check whether a word has a [p] in every third onset. This additional power comes from so-called fixed point operators, giving us $GP\nu$.

2.5 SPE

As was shown by Kaplan and Kay (1994), SPE can be implemented with finite state technology and generates regular languages, which is equivalent to weak Monadic Second-Order logic (MSO). Hence we can approximate SPE in MSO. That's good enough for us at this point.

2.6 Formal Hierarchy of Phonology

Mapping our linguistic theories to different logics, we obtain a surprising expressivity hierarchy with rich connections to formal language and automata theory on the one hand and algebra on the other.

	GP_0^0	GP_n^n	GP_n^ω	GP_ω^ω	GP_v/SPE
Modal logic	—	TL + NEXT	RTL	LTL/PLTL/NLTL	RLTL/ v -LTL
Classical logic	—	—	FO ²	FO ³	MSO
Formal language	—	—	—	star-free	regular
Algebra	—	—	locally 1-trivial	aperiodic	variety of all finite monoids
Automaton	—	—	—	counter-free	finite-state
Phenomenon	—	—	—	<i>nati</i>	stress assignment

Table 4: Hierarchy of classes of phonological theories

3 How Much Power do we Need?

3.1 The Power of Features

From a theorem of Thatcher (1967), it follows that all segment-based phonological theories, even GP_1^1 , have the same power as SPE, if we grant ourselves enough additional features to encode non-local dependencies. Details can be found in Kracht (1995a,b, 1997). This has two astonishing consequences:

(13) a. *Characterization of phonology*

While classes of syntactic theories can vary in their generative capacity (e.g. GPSG is strictly weaker than TAG), classes of phonological theories differ with regards to succinctness.

b. *Psycholinguistic indeterminacy*

As any complex segment-based phonological theory can be modeled with weaker operators using more features, classes of phonological theories make predictions concerning psycholinguistic aspects like parsing and learnability only with respect to specific sets of features.

Note that the inclusions in the phonological hierarchy are still proper if the set of features is fixed.

3.2 Beyond GP_n^ω — Unbounded Inbetweenness

n -Retroflexion in 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 ($/r/$, $/s/$) into a η iff the following conditions are fulfilled:

- (14) 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.

Theorem 3.1. *Nati can be accounted for in the class GP_ω^ω .*

Proof. We can translate the conditions above into three GP_ω^ω formulas as follows.

N1 *derived* $\eta \rightarrow \text{SINCE}(r \vee s, \neg \text{coronal} \wedge \neg n)$

N2 *derived* $\eta \rightarrow \triangleright \text{sonorant}$

N3 *derived* $\eta \rightarrow \neg \curvearrowright \text{retroflex continuant}$ □

Theorem 3.2. *The class GP_n^ω cannot model nati without the use of additional features.*

Proof. Since the formulas for (14b) and (14c) are RTL formulas, these conditions pose no problem for GP_n^ω . The culprit, then, has to be (14a). We know that theories in GP_n^ω can be formalized in FO^2 , the two-variable fragment of first order logic. But no FO^2 formula can impose restrictions on all nodes in an interval if there is no upper bound on the size of the interval. □

3.3 Beyond GP_ω^ω — Counting

Primary stress assignment in Creek and Cairene Arabic (Mitchell 1960; Haas 1977; Hayes 1995)

- (15) *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. Allegedly, there is no overt secondary stress.

Conditions (15a) and (15b) are trivial, but (15c) involves *modulo*-counting (e.g. $\text{mod } 2 = 0, 1, 0, 1, 0, 1 \dots$, $\text{mod } 4 = 0, 1, 2, 3, 0, 1, 2, 3 \dots$). Given (15d), we cannot use empirically untested features (like secondary stress) to assign primary stress without modulo counting.

Theorem 3.3. *Primary stress assignment in Cairene Arabic or Creek can be accounted for in the class SPE/GP_ν .*

Proof. It is easy to construct a finite state automaton for any stringset that involves counting *modulo* n , n finite. Finite state automata yield regular stringsets, which in turn are equivalent to the stringsets definable in MSO (Büchi 1960) or LTL with fixed point operators (Vardi 1988). □

Theorem 3.4. *The class GP_ω^ω cannot model primary stress assignment in Cairene Arabic or Creek without the use of additional features.*

Proof. It is well known that full first-order logic cannot do *modulo*-counting. As was proven 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 1991; Cohen et al. 1993), in which we formalized GP_ω^ω . □

4 Conclusion

- (16) *Accomplishments (in subjective order of importance)*
- Framework for study of different phonological theories
 - Phonological hierarchy
 - Characterization of phonology
 - Undefinability results
 - Discovery of locus of power (spreading)
 - Solution to theory internal problems (LCs)
 - Discovery of theory internal problems (Definition of spreading)

5 Appendix

5.1 Full Formalization

The material in this section is taken from Graf (2009).

Let E be some non-empty finite set of *basic elements* different from the neutral element ν (representing the empty set in GP's feature calculus). We define the set of *elements* $\mathcal{E} := (E \times \{1, 2\} \times \{\text{head}, \text{onset}\} \times \{\text{local}, \text{spread}\}) \cup (\{\nu\} \times \{1, 2\} \times \{\text{head}, \text{onset}\} \times \{\text{local}\})$. The set of melodic features $\mathcal{M} := \mathcal{E} \cup \{\mu, \text{fake}, \checkmark\}$ will be our set of propositional variables. We employ μ and \checkmark to mark unpronounced and licensed segments, respectively, and *fake* for unassociated onsets. For the sake of increased readability, the set of propositional variables is “sorted” such that $x \in \mathcal{M}$ is represented by m , $m \in \mathcal{E}$ by e , heads by h , 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 .

We furthermore use three nullary diamond operators, N , O , C , the set of which we designate by \mathcal{S} , read *skeleton*. In addition, we have two unary diamond operators \triangleleft and \triangleright , whose respective duals are denoted by \blacktriangleleft and \blacktriangleright . The set of well-formed formulas is built up in the usual way from \mathcal{M} , \mathcal{S} , \triangleleft , \triangleright , \rightarrow and \perp .

Our intended models $\mathfrak{M} := \langle \mathfrak{F}, V \rangle$ are built over bidirectional frames $\mathfrak{F} := \langle D, R_i, R_{\triangleleft} \rangle_{i \in \mathcal{S}}$, where $D \subseteq \mathbb{N}$, and $R_i \subseteq D$ for each $i \in \mathcal{S}$, and R_{\triangleleft} is the successor function over \mathbb{N} . The valuation function $V : \mathcal{M} \rightarrow \wp(D)$ maps propositional variables to subsets of D . The definition of satisfaction is standard.

$\mathfrak{M}, w \models \perp$	never
$\mathfrak{M}, w \models p$	iff $w \in V(p)$
$\mathfrak{M}, w \models \neg \phi$	iff $\mathfrak{M}, w \not\models \phi$
$\mathfrak{M}, w \models \phi \wedge \psi$	iff $\mathfrak{M}, w \models \phi$ and $\mathfrak{M}, w \models \psi$
$\mathfrak{M}, w \models N$	iff $w \in R_N$
$\mathfrak{M}, w \models O$	iff $w \in R_O$
$\mathfrak{M}, w \models C$	iff $w \in R_C$
$\mathfrak{M}, w \models \triangleleft \phi$	iff $\mathfrak{M}, w + 1 \models \phi$
$\mathfrak{M}, w \models \triangleright \phi$	iff $\mathfrak{M}, w - 1 \models \phi$

S1	$\bigwedge_{i \in \mathcal{S}} (i \leftrightarrow \bigwedge_{i \neq j \in \mathcal{S}} \neg j)$	Unique constituency
S2	$(\blacktriangleleft \perp \rightarrow O) \wedge (\blacktriangleright \perp \rightarrow N)$	Word edges
S3	$R \leftrightarrow (N \vee C)$	Definition of rhyme

S4	$N \rightarrow \triangleleft O \vee \triangleleft N$	Nucleus placement
S5	$O \rightarrow \neg \triangleleft O \vee \neg \triangleright O$	Binary branching onsets
S6	$R \rightarrow \neg \triangleleft R \vee \neg \triangleright R$	Binary branching rhymes
S7	$C \rightarrow \triangleleft N \wedge \triangleright O$	Coda placement

A propositional formula ϕ over a set of variables x_1, \dots, 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 $\neg x_i$. A phonological expression ϕ is an exhaustive propositional formula over \mathcal{E} such that $\phi \cup \{\mathbf{F1}, \mathbf{F2}, \mathbf{F3}, \bigvee h\}$ is consistent.

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

Let PH be the least set containing all such ϕ , and let $lic : PH \rightarrow \wp(PH)$ map every ϕ to its set of melodic licensors. By $S \subseteq PH$ we designate the set of phonological expressions occurring in magic licensing configurations (the letter S is mnemonic for “sibilants”). The following five axioms, then, sufficiently restrict the melody.

M1	$\bigwedge_{i \in \mathcal{S}} (i \rightarrow (\bigvee h_1 \wedge \bigvee o_1) \vee \mu \vee \text{fake})$	Universal annotation
M2	$((O \vee \triangleleft N \vee \triangleright N) \rightarrow \bigwedge \neg e_2)$	No pseudo branching for onsets, codas and branching nuclei
M3	$O \wedge \triangleleft O \rightarrow \bigwedge_{\phi \in PH} (\phi \rightarrow \bigvee_{\psi \in lic(\phi)} \triangleleft \psi)$	Licensing within branching onsets
M4	$C \wedge \bigwedge_{i \in \mathcal{S}} \neg i \rightarrow \triangleleft \neg \mu \wedge \bigwedge_{\phi \in PH} (\phi \rightarrow \bigvee_{\psi \in lic(\phi)} \triangleright \psi)$	Melodic coda licensing
M5	$\text{fake} \rightarrow O \wedge \bigwedge_{m \neq \text{fake}} \neg m$	Fake onsets

As mentioned above, we use μ to mark “mute” segments that will be realized as the empty string. The distribution of μ is simple for O and C : the former always allows it, the latter never does. For N , we first need to distribute \checkmark in a principled manner across the string to mark the licensed nuclei, which may remain unpronounced. Note that $\underline{\nu} \wedge \bar{\nu}$ by itself does not designate unpronounced segments (remember the phonological expression for $[\emptyset]$), and that unpronounced segments may not contain any other elements (which would affect spreading).

L1	$\mu \rightarrow \neg C \wedge \underline{\nu} \wedge \bar{\nu} \wedge (N \rightarrow \checkmark)$	Empty categories
L2	$N \wedge \triangleleft N \rightarrow (\mu \leftrightarrow \triangleleft \mu)$	Licensing of branching nuclei
L3	$O \wedge \triangleleft O \rightarrow \neg \triangleleft \mu \wedge \neg \mu \wedge \neg \triangleright \mu$	Licensing of branching onsets
L4	$N \wedge \checkmark \leftrightarrow \underbrace{\triangleright (C \wedge \bigvee_{i \in \mathcal{S}} i)}_{\text{Magic Licensing}} \vee \underbrace{(\neg \triangleleft N \wedge \blacktriangleright \perp)}_{\text{FEN}} \vee$ $\underbrace{((\neg \triangleleft N \rightarrow \triangleleft (\triangleleft N \vee \blacktriangleleft \perp)) \wedge (\neg \triangleright N \rightarrow \triangleright \triangleright (N \wedge \neg \mu)))}_{\text{Proper Government}}$	P-licensing

We define a general spreading scheme σ with four parameters i , j , ω and ϖ .

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

The variables $i, j \in \mathcal{E}$, coupled with judicious use of the formulas ω and ϖ regulate the optionality of spreading. In the first case, i is a spread element and ω, ϖ 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, ϖ describe the source and the set of targets. Note that we need to make sure that every structural configuration is covered by some ω , so that unwanted spreading can be blocked by making ϖ not satisfiable. As further parameters, the finite values $min, max > 0$ encode the minimum and maximum distance of spreading, respectively. Finally, the operator $\diamond \in \{<, >\}$ fixes the direction of spreading for the entire formula (\diamond^n is the n -fold iteration of \diamond , of course). With optional spreading, the direction of the operator is opposite to the direction of spreading, otherwise they are identical.

5.2 In how far is MTP Different from Other Approaches in Computational Phonology?

First, MTP allows for succinct, explicit descriptions. Compare the formulas in Figure 9 on the following page to their respective automata. In particular, it is easy to see what structures the conjunction of both formulas yields, but intersecting the automata obfuscates their content, and they have to make reference to the remaining constituent features and thus are less general than the formulas, which ignore all but the essential parts of the structure. Moreover, research in computational phonology mostly ignores results on subregular languages and restricts itself to finite-state implementability, which is too coarse a notion for an insightful study of phonology. Furthermore, the automata perspective treats theories as monolithic entities, whence abstraction and modularization are difficult. In most cases, it is even necessary to modify the theory to a notable degree to make it implementable (e.g. compressing feature bundles into single digits representing them). Finally, classical computational approaches are restricted to results of weak generative capacity, whereas the logical approach talks directly about structures and many of its results can be lifted easily to trees and graphs in general.

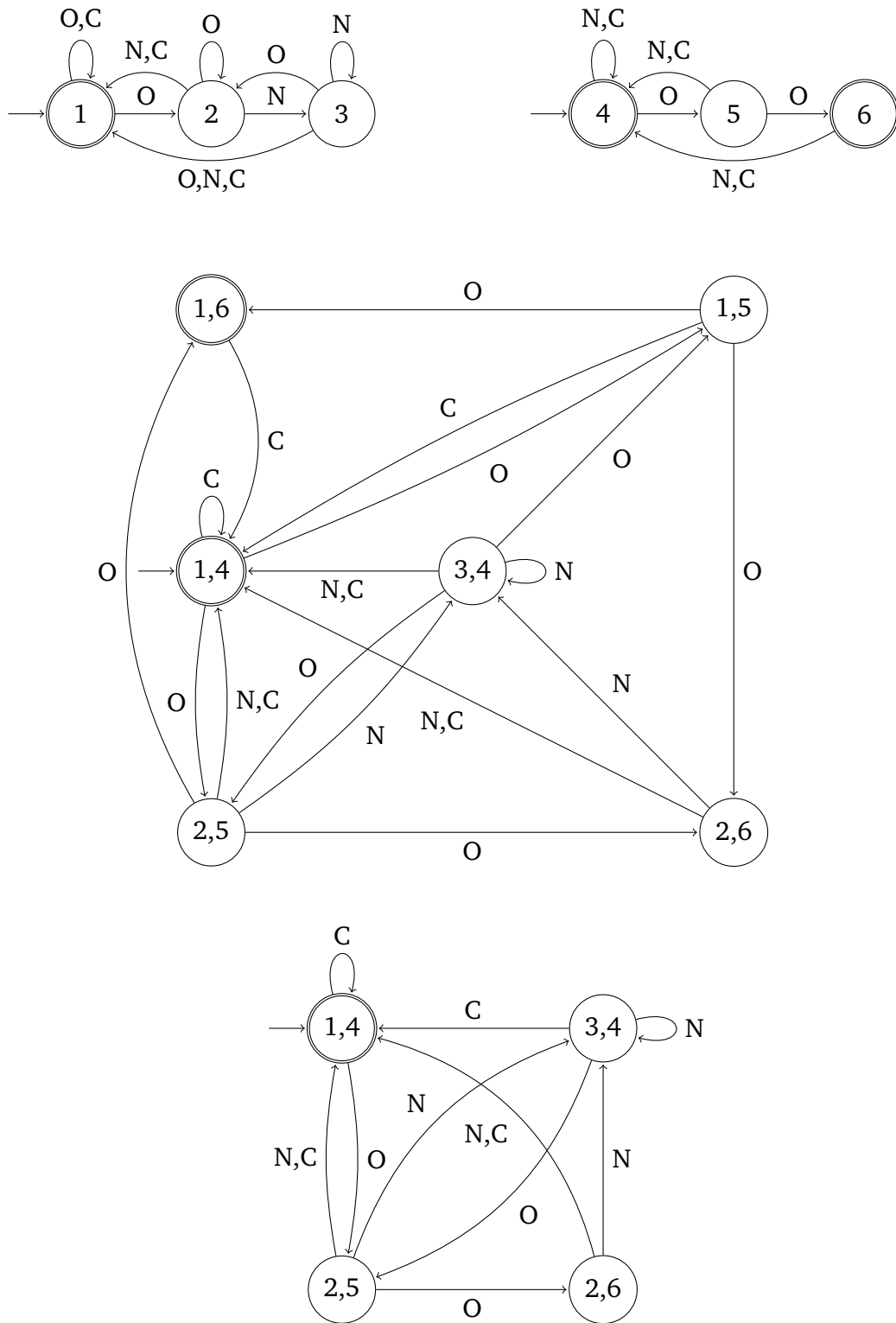


Figure 9: Automata for $N \rightarrow \triangleleft O \vee \triangleleft N$ and $O \rightarrow \neg \triangleleft O \vee \neg \triangleright O$, and their intersection (inaccessible states removed), plus the minimal automaton for their intersection

References

- Backofen, Rolf, James Rogers, and K. Vijay-Shanker. 1995. A first-order axiomatization of the theory of finite trees. Technical Report RR-95-05, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Erwin-Schrödinger Straße, Postfach 2080, 67608 Kaiserslautern, Germany.
- Blackburn, Patrick. 1993. Modal logic and attribute value structures. In *Diamonds and defaults*, ed. M. de Rijke, 19–65. Norwell, MA, USA: Kluwer Academic Publishers.
- Blackburn, Patrick. 1994. Structure, languages and translations: the structural approach to feature logic. In *Constraints, language and computation*, ed. C. Rupp, M. Rosner, and R. Johnson, 1–27. San Diego, CA: Academic Press.
- Blackburn, Patrick, Claire Gardent, and Wilfried Meyer-Viol. 1993. Talking about trees. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, 30–36.
- Blackburn, Patrick, and Wilfried Meyer-Viol. 1994. Linguistics, logics, and finite trees. *Logic Journal of the IGPL* 2:3–29.
- Blackburn, Patrick, and Edith Spaan. 1993. A modal perspective on the computational complexity of attribute value grammar. *Journal of Logic, Language and Information* 2:129–169.
- 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.
- Chomsky, Noam. 1986. *Barriers*. Cambridge, Mass.: MIT Press.
- 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.
- Graf, Thomas. 2009. Towards a factorization of string-based phonology. Paper submitted to the ESLLI 2009 Student Session.
- 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).
- Hansson, Gunnar Ólafur. 2001. *Theoretical and typological issues in consonant harmony*. Doctoral Dissertation, University of California, Berkeley.
- Hayes, Bruce. 1995. *Metrical stress theory*. Chicago: Chicago University Press.
- Kaplan, Ronald M., and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378.
- 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.
- Kracht, Marcus. 1995a. Is there a genuine modal perspective on feature structures? *Linguistics and Philosophy* 18:401–458.
- Kracht, Marcus. 1995b. Syntactic codes and grammar refinement. *Journal of Logic, Language and Information* 4:41–60.
- Kracht, Marcus. 1997. Inessential features. In *Logical aspects of computational linguistics*, ed. Alain Lecomte, F. Lamarche, and G. Perrier. Berlin: Springer.
- Macdonell, Arthur. 1910. *Vedic grammar*. Strassburg: Trübner.
- McNaughton, R., and S. Pappert. 1971. *Counter-free automata*. Cambridge, Mass.: MIT Press.
- Mitchell, Terence F. 1960. Prominence and syllabification in Arabic. *Bulletin of the School of Oriental and African Studies* 23:369–389.
- Potts, Christopher, and Geoffrey K. Pullum. 2002. Model theory and the content of OT constraints. *Phonology* 19:361–393.
- Rogers, James. 1997a. “Grammarless” phrase structure grammar. *Linguistics and Philosophy* 20:721–746.

- Rogers, James. 1997b. A unified notion of derived and derivation structures in TAG. In *Proceedings of the Fifth Meeting on Mathematics of Language MOL5 '97*. Saarbrücken.
- Rogers, James. 1998a. *A descriptive approach to language-theoretic complexity*. Stanford: CSLI.
- Rogers, James. 1998b. A descriptive characterization of tree-adjoining languages. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING'98) and the 36th Annual Meeting of the Association for Computational Linguistics (ACL'98)*. Montreal.
- Rogers, James. 1998c. On defining TALs with logical constraints. In *Fourth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+4)*, ed. Anne Abeillé, Tilman Becker, Owen Rambow, Giorgio Satta, and K. Vijay-Shanker, 151–154.
- Rogers, James. 2003. Syntactic structures as multi-dimensional trees. *Research on Language and Computation* 1:265–305.
- Schein, Barry, and Donca Steriade. 1986. On geminates. *Linguistic Inquiry* 17:691–744.
- Stabler, Edward P. 1992. *The logical approach to syntax: Foundations, specifications and implementations of theories of government and binding*. Cambridge, Mass.: MIT Press.
- Sturtevant, William C., ed. 1987. *A Creek source book*. New York: Garland.
- Thatcher, J. 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.
- Vardi, Moshe Y. 1988. A temporal fixpoint calculus. In *Proceedings of the 15th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, 250–259.
- Whitney, William D. 1889. *Sanskrit grammar*. London: Oxford University Press.