A Model Theoretic Perspective on Phonological Feature Systems

Scott Nelson
Stony Brook University
Department of Linguistics
Institute for Advanced Computational Science
scott.nelson@stonybrook.edu

Abstract

This paper uses model theory to analyze the formal properties of three phonological feature systems: privative, full binary, and underspecified. By systematically manipulating the choice of logical language and representational primitives, it is shown that logical negation effectively converts any feature system into a full binary one. This further implies that in order to have underspecification or non-binary feature oppositions, valuation should be encoded into the representational primitives rather than derived through the logical connectives. These results are obtained by comparing the predicted natural classes of each formalization.

1 Introduction

Phonological features are present in some form in most modern theories of phonology. While there are debates over how to best represent features, it is typically agreed that features encode sub-segmental acoustic and/or articulatory properties. A feature system is a set of valued features where the valuation is typically drawn from the set $\{+, -\}$. Segments are therefore shorthand for collections of valued features, and rules or constraints use features to target groups of sounds that undergo the same phonological processes.

In practice, feature systems also regularly contain a 0 valuation to imply that a certain segment is not specified as either $+$ or $-$ for a given feature. The 0 valuation seems to serve two theoretical purposes. The first purpose is simply as a placeholder for a feature that does not apply for a given segment. For example, the feature [distributed] differentiates between coronal segments made with the tip or blade of the tongue. For non-coronal sounds, this distinction is meaningless and therefore is usually represented with 0. The second purpose that 0 serves is for underspecification. Feature systems that use underspecification do so in order to ensure that certain rules do not target specific segments, even though they share a certain phonetic property. For example, sonorant sounds are phonetically voiced, but in some feature systems they are phonologically underspecified as [0 voice] which allows them to be excluded from phonological voicing assimilation processes.

Most feature systems mix $\{+, -, 0\}$ in different ways, but it is not clear whether or not each system can be formally represented and interpreted in the same way. It is also worth considering whether or not each feature system is meaningfully different than the others, or if it can be thought of as a notational variant. One set of tools that allows for looking into these types of questions is model theory and logic. Model theory is a branch of mathematics that allows for the precise definition of relational structures such as strings (Libkin, 2004). These structures can be further evaluated using different types of logic.

Model theory and logic can therefore provide a meta-language to compare different types of phonological representations. Strother-Garcia (2019) compares different types of syllabic representations, Jardine et al. (2021) study the difference between traditional autosegmental representations (Goldsmith, 1976) and Q-theoretic representations (Inkelas and Shih, 2016), and Oakden (2020) shows how different representations of tone are essentially notational variants. Another advantage of using model theory for phonology is that it has a well defined relationship with computational complexity and learnability (Strother-Garcia et al., 2016; Vu et al., 2018; Chandlee et al., 2019). Additionally, it provides a way to formalize differences between representational structures so that we can move beyond relying solely on our intuitions. Phonological feature systems are one area that has yet to be explored in this way.

In this paper I will use model theory and logic

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 1-10.
Held on-line February 7-9, 2022

to explore three types of feature systems: a privative system that uses $\{+, 0\}$, a full binary system which uses $\{+, -\}$, and a contrastive system that uses $\{+, -, 0\}$. While Mayer and Daland (2020) provide different algorithms for a how these feature systems could be learned, this paper focuses on how each feature system can be formally represented using different types of logics and representational primitives. The diagnostic that I will focus on are the natural classes that are expected for a given feature system.

Previous mathematical treatments of feature systems have primarily focused on the binary aspect of features (Bale and Reiss, 2018; Keenan and Moss, 2009; Johnson, 1972). Their systems look like the full binary system where every segment is specified as either $+$ or $-$ for every feature. The reason for this is due to their use of logical negation. The main result I will show is that a full binary feature system is the only possible result when using logical negation. Consequently, in order to effectively have 0 values, the positive/negative feature valuation must be encoded into the representational primitives rather than emerge from the logical connectives.

2 Phonological Features

The use of phonological features as a tool for phonological analysis is typically traced back to the Prague School, notably Nikolai Trubetzkoy and Roman Jakobson. Trubetzkoy (1939) proposed three different types of feature based oppositions: privative, gradual, and equipollent. A privative feature in his analysis would be [voice] where a segment either has the property of being voiced or it does not. Gradual features are things such as [height 1], [height 2], ..., [height n] where the numerical valuations encode the vowel height scale. Equipollent features are similar to privative features in that they are present or absent, but unlike privative features they do not encode a binary-like opposition. The examples of features used to explain an equipollent opposition are [labial], [coronal], and [dorsal].

Jakobson's contribution to feature theory culminated with *Preliminaries to Speech Analysis* (Jakobson et al., 1951). In this monograph, all features were treated as binary, specifically encoded as being either $+$ and $-$ for each feature. The use of binary features in phonology was further amplified due to their inclusion in *The Sound Pattern of English* (Chomsky and Halle, 1968) and they continue to be used as the default valuation of features in most modern phonology textbooks.

As feature theory has developed over the last decade, there have been debates along multiple dimensions about how best to represent features. One dimension is whether or not features should be thought of as attributes or particles in the terms of Ladd (2014). That is, should we think about features in terms of feature bundles that are ordered temporally, or should we think about them in autosegmental terms where each feature is specified on its own tier and has some type of relation to a general timing unit. In this paper I will focus on the former as it is more typical. Nonetheless, the results of this study should be able to be generalized to autosegmental or feature geometric systems (McCarthy, 1988).

A second dimension in the debate on features has to do with whether or not features should be thought of as discrete or gradient categories. The gradient approach often is lumped in with a scalar approach (e.g., Flemming, 2001), but it is possible to have scalar features without forgoing discrete categories. Since I am using finite model theory in this paper, the feature set needs to be finite and therefore discrete categories are necessary. However, it is also possible to approximate gradient feature values by having a large, but finite, set of possible numerical valuations.

Two other debates have to do with whether or not features are innate or emergent (Mielke, 2008), or whether or not features contain phonetic substance or instead are substance free (**?**). Neither of these two areas directly affect feature valuation and will be left aside.

2.1 Natural Classes

Natural classes are the result of partitioning a language's segment inventory using phonological features. Traditionally, there are two explanations for natural classes. The phonetic explanation is that all segments that form a natural class share one or more phonetic property. The distributional explanation is that all segments that form a natural class are the target/trigger for a phonological process or involved in some type of constraint. One problem with these explanations is that they do not always cohere (Duanmu, 2016). For example, there are groups of segments that have the same distribution, but nonetheless do not share a phonetic property. Mielke (2008, p. 12) attempts to explain this dis-

connect by arguing for emergent features. He also offers the following definition that will be useful for current purposes: a natural class is, "a group of sounds in an inventory which share one or more distinctive features, *within a particular feature theory* to the exclusion of all other sounds in the inventory," (emphasis original).

It also should be clarified what it means for two segments to share a feature. As Bale and Reiss (2018) point out, phonologists can be sloppy when talking about features by not clearly distinguishing the difference between a feature and a segment's specific valuation for a feature. It is safe to assume that what Mielke means in the quote above is that two segments are part of the same natural class when they share the same *valuation* for one or more distinctive features. In set theoretic terms we can think of natural classes as groups of sounds whose valued feature intersection is non-empty.

Conjunction would therefore be the logical parallel to this and conjunction does seem to be the way in which phonologists tend to think about natural class formation. For example, Kenstowicz and Kisseberth (1979, p. 241) write, "...an adequate feature system should permit any natural class of sounds to be represented by the conjunction of features in a matrix," while Odden (2005, p. 49) writes, "Natural classes can be defined in terms of conjunctions of features...." If a phoneme /n/ had the feature bundle [+coronal +sonorant -continuant +nasal], then we say that /n/ has the properties of being +coronal AND + sonorant AND -continuant AND +nasal.

While conjunction is the main way in which features seem to be combined to form natural classes, there are other possible logical connectives that one might use. For example, the curly brackets {} were used by Chomsky and Halle (1968) to indicate disjunctive triggering environments. Furthermore, Mielke (2008) showed that ∼97% of the phonologically active classes he looked at can be described with the *SPE* feature system if disjunction is allowed. This is an increase of 26% from *SPE*'s coverage without disjunction, which seems like a positive finding, but if we abstract away from the specific features used in any given system, disjunction should be able to cover 100% of natural classes.[1] One reason why we may not want disjunction, despite its ability to allow for broad empirical coverage, is the fact that with arbitrary disjunction any subset of segments can form a natural class.

While logical negation can be interpreted as complementation, a reviewer points out that its use for defining natural classes has largely been avoided (e.g. Chomsky and Halle (1968)). A notable exception is Hayes and Wilson (2008) which employed a complementation operator in their definition of constraints.

Quantification is another tool used in formal logic that could be used for interpreting feature bundles. For the most part, phonologists seem to stay away from quantification, but Reiss (2003) uses it to define identity relations in the structural description of phonological rules. Since the structural description is usually thought to be a natural class, this could be one area where quantification is used for interpreting feature bundles. That being said, identity is often baked into the axioms of logical interpretation languages. Strother-Garcia (2019) discusses the relationship between quantifier-free logics and locality for syllabification, but it is worth pursuing whether or not this is the right approach when considering phonological features. This is left for future work.

2.2 Underspecification

0 values are often associated with underspecification. Underspecification is when certain features are not assigned either a plus or a minus value for a given feature. Two common types of underspecification are privative underspecification where minus values are completely eliminated and only + feature values are assigned, and contrastive underspecification where any feature value that is redundant is removed. For example, sonorants can have a 0 value for the [voice] feature since sonorants do not have a voicing contrast and are by default [+voice]. The redundant value for [voice] is then filled back in at the end of the derivation. Sonorants being underspecified for [voice] has been a central argument in the debate around contrastive underspecification and will be used in the current analysis as well (see Steriade (1995) for further discussion and review).

0 values can also be used for non-redundant purposes. This is sometimes used when a feature only applies to a certain class of sounds (Hayes, 2011). Steriade (1995, p. 117) calls this "trivial" underspecification, contrasting it with the "temporary" underspecification described in the previous para-

[1] The reason that *SPE* (and the other feature systems evaluated by Mielke (2008)) do not reach 100% is because they are unable to contrast between certain types of segments such as pre-nasalized/post-nasalized stops.

graph. In the analysis in this paper, the distinction between trivial and temporary underspecification collapses because only the natural classes the phonological feature matrices represent is under consideration.

3 Model Theory

3.1 String Models

Strings can be straightforwardly defined in model theory. At minimum, a model theoretic representation includes a finite domain $\mathcal{D}$ and a finite set of relations $\mathcal{R}$. $\mathcal{R}$ also typically includes a set of labeling relations drawn from a primitive set of symbols Σ onto elements of the domain, and an ordering relation used to structure the domain elements. Σ is typically referred to as the alphabet since it contains the segmental labels for the domain elements. I will use the successor ordering relation throughout this paper. The domain is typically taken to be the natural numbers $\mathbb{N}$. Given this, we can define successor as $\langle i, i+1\rangle \in \mathcal{D} \times \mathcal{D}$. A model signature is a tuple containing all of this information. For the successor model $\mathcal{M}^{\lhd}$, this contains $\langle \mathcal{D}, \mathcal{R}_\sigma | \sigma \in \Sigma, \lhd \rangle$.

$$\mathcal{D} = \{1, 2\}$$
$$\mathcal{R}_a = \{2\}$$
$$\mathcal{R}_b = \{1\}$$
$$\lhd = \{\langle 1, 2\rangle\}$$

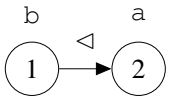

Figure 1: Successor word model for $\mathcal{M}^{\lhd}_{ba}$.

Figure 1 shows the successor word model for the word ba given the alphabet $\Sigma = \{a, b\}$. This defines the word over segments. As phonologists we may want to analyze this structure using features, but since features are not innate to the model, we have to define them ourselves. One way to do this is with user defined predicates. These are predicates that the analyst imposes on the model. Features can be defined disjunctively from a segment based model such as the one in Figure 1. For example, we define the predicate `voi` as:

(1) $\mathsf{voi}(x) \stackrel{\text{def}}{=} \mathcal{R}_a(x) \vee \mathcal{R}_b(x)$

This formula says that any segment that is labeled as a or b has the property of being voiced. Features are therefore epiphenomenal in this type of model.

A second option is to have our alphabet Σ be made of phonological features rather than phonological segments. This also requires a change to the labeling relations. Typically, each domain element is given a single label. If phonological features are the primitives, then it must be the case that a single domain element can have more than one label. Figure 2 shows a second successor model for the word ba, this time using features as the alphabetic primitives rather than segments. With this type of model we can define segments conjunctively using features. In this case, it is the segments which are epiphenomenal.

$$\mathcal{D} = \{1, 2\}$$
$$\mathcal{R}_{\texttt{+voi}} = \{1, 2\}$$
$$\mathcal{R}_{\texttt{+lab}} = \{1\}$$
$$\mathcal{R}_{\texttt{-cont}} = \{1\}$$
$$\mathcal{R}_{\texttt{-syl}} = \{1\}$$
$$\mathcal{R}_{\texttt{+syl}} = \{2\}$$
$$\mathcal{R}_{\texttt{+back}} = \{2\}$$
$$\mathcal{R}_{\texttt{-high}} = \{2\}$$
$$\lhd = \{\langle 1, 2\rangle\}$$

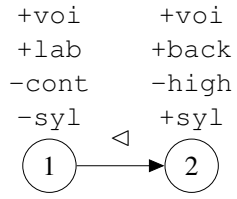

Figure 2: Successor with features model for *ba*.

In the example given here, *valued* features make up the primitive units. That is, $+$ and $-$ values are built directly into the individual labeling relations. Another possibility would be to only have feature labels as the set of primitives and interpret feature valuation as whether or not a domain element has a given feature label. For example, a [+voice] domain element would be one that is labeled with the feature [voice] while a [-voice] domain element would be one that does not have the label [voice]. I will refer to the first style, where the $+/-$ values are encoded directly into the primitives, as *bivalent primitives*. The second style, where only a feature itself is encoded into the primitives, will be referred to as *univalent primitives*.

3.2 Logical Evaluation

Model theoretic structures can be interpreted with different types of logic. First-order logic (FO) is commonly used, but it allows for quantification which does not seem to be used when describing phonological natural classes. Quantifier-free logic (QF) is like FO except without quantifiers. Even this is likely too powerful since it still uses standard logical connectives like conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$), and implication ($\rightarrow$). Conjunction and possibly negation are the only primitives that seem to be required for defining natural classes and yet if they are both allowed to

freely combine we can then derive the other logical connectives. For example, disjunction $(P \vee Q)$ can be defined as $\neg(\neg P \wedge \neg Q)$. One solution to this problem is to restrict the use of negation to only atomic sentences. We can do so by defining different types of sub-QF logics.

Of the three logics we can define this way, two of them will be used in this paper. Conjunction of negative and positive literals (CNPL) allows for the conjunction of any sentence within the language, but negation is only allowed to scope over atomic predicates. Conjunction of positive literals (CPL) only allows for the conjunction of sentences. Negation is strictly excluded from the logical language. Conjunction of negative literals (CNL) is the third logical language and only allows for negated atomic primitives to be combined with conjunction. The syntax of CNPL and CPL are recursively defined in (2).

(2) (a) CNPL
  i. Base case: For all atoms P, "P" and "$\neg P$" are sentences.
  ii. Inductive case: For all sentences A, B, "$A \wedge B$" is a sentence.

(b) CPL
  i. Base case: For all atoms, P, "P" is a sentence.
  ii. Inductive case: For all sentences A, B, "$A \wedge B$" is a sentence.

For this paper, the atoms are the feature labeling relations.

4 Model Theoretic Feature Systems

In this section I will demonstrate how different phonological feature systems can be expressed using model theory. The diagnostic used in this analysis is a comparison of the natural classes that a certain feature system is predicted to have based on a feature table versus what type of natural classes can be formed from the model theoretic representation and interpretation. Phonological feature systems are typically presented as tables of $+$, $-$, and 0 values. Three examples are shown in Table 1 (adapted from Mayer and Daland (2020)).

The privative feature system uses only $+$ and 0, the full binary system uses only $+$ and $-$, and the contrastive system uses $+$, $-$, and 0. Each of these therefore predicts different sets of natural classes. Since the 0 value typically represents the lack of a valuation, the natural classes for each feature system are based on similarity of $+$ and $-$ values. The set of natural classes for the privative feature system is therefore {{N},{N,D}}. There are in fact two ways to define the subset {N} in this feature system: segments that are [+son] or segments that are [+son, +voi]. The subset {N,D} is defined as all segments that are [+voi]. The set of natural classes for the full system is {{N}, {N,D}, {D}, {T}, {D,T}} and the set for the contrastive system is {{N}, {D}, {T}, {D,T}}. Construction of these sets was done the same way as described for the privative feature system.

	Privative		Full		Contrastive	
	son	voice	son	voice	son	voice
N	+	+	+	+	+	0
D	0	+	-	+	-	+
T	0	0	-	-	-	-

Table 1: Example of three types of feature systems. N represents all sonorants, D represents voiced obstruents, and T represents voiceless obstruents.

There are two reasonable ways in which we can turn these feature tables into model theoretic representations. The first way would be to use a segmental model and define translations from the segmental model into different feature models. MSO-definable string to string transformations (Courcelle, 1994; Engelfriet and Hoogeboom, 2001) allow for translation between different representational systems. A second way would be to use the feature successor model and have the difference in valuations emerge from the definitions of each specific model. Both methods will result in the same structures for evaluation, but I will take the second approach as it aligns more directly with the theme and discussion of the paper so far.

The primary model signature that will be used is the successor model defined above: $\mathcal{M}^{\triangleleft} = \langle \mathcal{D}, \mathcal{R}_\sigma | \sigma \in \Sigma, \triangleleft \rangle$. We can alter the general successor model slightly by providing fixed labeling relations. This allows for the definition of two model signatures: a univalent primitive signature $\mathcal{M}^{\upsilon}$ and a bivalent primitive signature $\mathcal{M}^{\beta}$. These are defined as follows:

(3) $\mathcal{M}^{\upsilon} = \langle \mathcal{D}, \texttt{voi}, \texttt{son}, \triangleleft \rangle$

(4) $\mathcal{M}^{\beta} = \langle \mathcal{D}, +\texttt{voi}, +\texttt{son}, -\texttt{voi}, -\texttt{son}, \triangleleft \rangle$

We can further specify models for each feature system (privative = P, full = F, contrastive = C).

This leaves us with six potential structures: $\mathcal{M}^{\upsilon}_{\text{P}}$, $\mathcal{M}^{\upsilon}_{\text{F}}$, $\mathcal{M}^{\upsilon}_{\text{C}}$, $\mathcal{M}^{\beta}_{\text{P}}$, $\mathcal{M}^{\beta}_{\text{F}}$, and $\mathcal{M}^{\beta}_{\text{C}}$. Each can then be evaluated using CPL and CNPL.

Since these models define strings, I will define the string DNT.[2] For the univalent primitive signature ($\mathcal{M}^{\upsilon}$), I will assume that any segment with a $+$ value in the feature table will be labeled with that feature. In this case, both 0 and $-$ values do not correspond to a label. For the bivalent primitive signature ($\mathcal{M}^{\beta}$), I will assume that any segment with a $+$ value in the feature table will be assigned the $+f$ label and any segment with a $-$ value in the feature table will be assigned the $-f$ label. The 0 once again will correspond to no label.

4.1 Evaluating Univalent Primitive Models

Let us start by looking at the different univalent feature models as interpreted with CPL logic. As it turns out, the privative and full features systems have an identical structure under $\mathcal{M}^{\upsilon}$. This is not all that surprising since a privative model just replaces all of the $-$ values with 0 values. In other words, both types of feature system allow for binary distinctions to be made, but the full feature system does it explicitly with a $-$ while the privative system does it through presence/absence of a feature. The top of Figure 3 shows the model for the string DNT.

As can be seen, domain element 1 which corresponds to D is only labeled with the `voi` label while domain element 2 which corresponds to N is labeled with both the `voi` and `son` labels. Domain element 3 is left unlabeled since T has no corresponding $+$ features in either the privative or full feature charts. The model for the contrastive feature system is shown in the bottom of Figure 3.[3] It differs slightly from the first model signature due to N having a 0 value for `voi` since voicing is not contrastive for sonorants in this feature system. Because of this, domain element 2 only receives the `son` label.

Given these model theoretic structures, we can now interpret them logically. Since our first evaluation logic is CPL, we can look at which domain elements satisfy all of the predicates we can make using conjunction over positive literals. The primitives are the features `voi` and `son`, so there are three predicates: the singletons `voi` and `son`, as well as the conjunction of the two `voi` $\wedge$ `son`. Table 2 shows the resulting classes of sounds from interpreting the structures in this way.

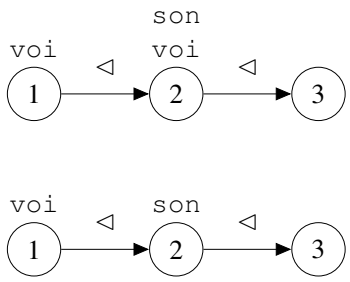

Figure 3: Models for the string DNT using models $\mathcal{M}^{\upsilon}_{\text{P}} = \mathcal{M}^{\upsilon}_{\text{F}}$ (top) and $\mathcal{M}^{\upsilon}_{\text{C}}$ (bottom)

CPL($\mathcal{M}^{\upsilon}$)	$\mathcal{M}^{\upsilon}_{\text{P}}$	$\mathcal{M}^{\upsilon}_{\text{F}}$	$\mathcal{M}^{\upsilon}_{\text{C}}$
`voi`	{N,D}	{N,D}	{D}
`son`	{N}	{N}	{N}
`son` $\wedge$ `voi`	{N}	{N}	{}
MISSING	–	{D}, {T}, {D,T}	{T}, {D,T}
EXTRA	–	–	–

Table 2: CPL logical interpretation of the different univalent primitive model theoretic structures.

The classes for $\mathcal{M}^{\upsilon}_{\text{P}}$ and $\mathcal{M}^{\upsilon}_{\text{F}}$ are {{N},{N,D}}. For $\mathcal{M}^{\upsilon}_{\text{P}}$, which is the correlate of the privative feature system, this is the expected result. That is, it matches the set of natural classes that we would predict given the feature table in Table 1. For $\mathcal{M}^{\upsilon}_{\text{F}}$, this is an under-prediction. As can be seen in the MISSING row of Table 2, $\mathcal{M}^{\upsilon}_{\text{F}}$ as interpreted with CPL fails to generate the classes {{D},{T},{D,T}} which a full binary feature system should have. The reason these classes are not generated is because they require an ability to reference minus values in some way. This is not possible given the CPL with univalent primitive system used here. $\mathcal{M}^{\upsilon}_{\text{C}}$ as interpreted with CPL correctly rules out the class {D,N}, which is one of the primary goals of the contrastive feature system, but still under predicts in a similar way to the full model. This once again has to do with not being able to reference minus values for natural class formation.

Overall, CPL logic with univalent primitives is a good way of representing privative feature systems since the lack of minus features aligns with CPL's inability to target minus features. For the other two feature systems, we need to be able to reference minus features in order to obtain the desired natural classes. One way that this may be accomplished

[2] Since N indicates all sonorant sounds this could correspond to words like *bus* or *juice*.

[3] I will only show the visual representation of models in the main body of the paper from here on out.

is through the use of negation. As mentioned in a previous section, we want to limit our negation to the atomic elements, which in this case are feature values. This allows for a straightforward interpretation such that atomic elements on their own can be thought of as $+F$ for some atomic feature and negated atomic elements can be thought of $-F$ for the same feature. CNPL as our interpretation logic allows us to take this route.

CNPL($\mathcal{M}^v$)	$\mathcal{M}^v_{\text{P}}$	$\mathcal{M}^v_{\text{F}}$	$\mathcal{M}^v_{\text{C}}$
voi	{N,D}	{N,D}	{D}
¬voi	{T}	{T}	{N,T}
son	{N}	{N}	{N}
¬son	{D,T}	{D,T}	{D,T}
son ∧ ¬son	{}	{}	{}
son ∧ voi	{N}	{N}	{}
son ∧ ¬voi	{}	{}	{N}
¬son ∧ voi	{D}	{D}	{D}
¬son ∧ ¬voi	{T}	{T}	{T}
voi ∧ ¬voi	{}	{}	{}
MISSING	–	–	–
EXTRA	{D}, {T}, {D,T}	–	{N,T}

Table 3: CNPL logical interpretation of the different univalent primitive model theoretic structures.

Table 3 shows the interpretation of the $\mathcal{M}^v$ structures using CNPL logic. Once again the privative and full feature system models will have the same set of classes: {{N}, {N,D}, {D}, {T}, {D,T}}. In this case, this is the set of classes that we would expect for the full feature system. This means that the privative model now overpredicts in regards to natural class formation. As can be seen in the EXTRA row of Table 3, the classes {{D}, {T}, {D,T}} are generated because the use of negation effectively turns every feature into a binary feature. For the contrastive feature system, this also presents a problem. In the contrastive system, a distinction needs to be made between the negative value for a feature and the lack of any value for a feature. Logical negation collapses this distinction. As can be seen in the third column, $\mathcal{M}^v_c$ considers N to be part of the ¬voi class. So not only does CNPL with univalent features over predict in the case of the contrastive feature system model, it over predicts by creating a class that none of the three feature systems uses.

A univalent model interpreted with CNPL therefore best models a full feature system where every segment is fully specified for either + or −. Since the privative and full feature systems have the same model signature in this analysis, the meaningful difference between these two systems seems to be in how the structures are interpreted logically rather than how the structures are labeled.[4] It also appears that there is no way to accurately represent a contrastive feature system with univalent primitives using either of the two interpretation logics. For contrastive feature systems it is necessary to target minus feature values when defining natural classes, but it is also necessary to maintain the distinction between a 0 value and a − value. One way in which this may be accomplished is to strictly encode the feature valuation into the primitives rather than using logical negation to explain the $+/-$ distinction.

4.2 Evaluating Bivalent Primitive Models

Figure 4 shows the models for $\mathcal{M}^\beta_{\text{P}}$, $\mathcal{M}^\beta_{\text{F}}$, and $\mathcal{M}^\beta_{\text{C}}$. Recall that in $\mathcal{M}^\beta$, the primitives include $+$voi, $+$son, $-$voi, and $-$son. Each of the three model signatures varies in how much information is encoded. For all models, a + value for a feature results in a label of $+F$ and a − value for a feature results in a label of $-F$. Unlike the univalent models, each feature system here does result in a unique model theoretic structure. This means that the difference between the feature systems cannot be explained by the logical interpretation language.

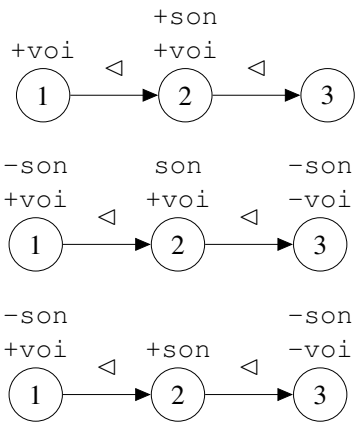

Figure 4: Models for the string DNT using models $\mathcal{M}^\beta_{\text{P}}$ (top), $\mathcal{M}^\beta_{\text{F}}$ (middle), and $\mathcal{M}^\beta_{\text{C}}$ (bottom).

Given these structures, we can once again interpret them logically using CPL. Table 4 shows what

[4] If we were to use negative valued feature labels as our univalent primitives this distinction may not hold. In this case, all segments would be unlabeled for the privative feature system. So it is not necessarily any univalent feature model where the distinction between privative and full feature systems is in the logical interpretation, but rather a univalent feature model that encodes positive feature values.

classes result from the models. Notice that each model does not contain any extra, nor have any missing, classes. As it turns out, the interpretation of each model now results in the exact set of natural classes that the corresponding feature set predicts.

CPL($\mathcal{M}^\beta$)	$\mathcal{M}^\beta_{\text{P}}$	$\mathcal{M}^\beta_{\text{F}}$	$\mathcal{M}^\beta_{\text{C}}$
`+voi`	{N,D}	{N,D}	{D}
`−voi`	{}	{T}	{T}
`+son`	{N}	{N}	{N}
`−son`	{}	{D,T}	{D,T}
`+son` $\wedge$ `−son`	{}	{}	{}
`+son` $\wedge$ `+voi`	{N}	{N}	{}
`+son` $\wedge$ `−voi`	{}	{}	{}
`−son` $\wedge$ `+voi`	{}	{D}	{D}
`−son` $\wedge$ `−voi`	{}	{T}	{T}
`+voi` $\wedge$ `−voi`	{}	{}	{}
MISSING	–	–	–
EXTRA	–	–	–

Table 4: CPL logical interpretation of the different bivalent primitive model theoretic structures.

Given that feature bundles are interpreted conjunctively and we used a logic that only contains conjunction, this result is not all that surprising. That being said, the same logic was used to interpret the model theoretic structures defined with univalent primitives and there was only able to capture a privative feature system. This highlights the interaction between representation and logical interpretation. Depending on the representations used, different logics result in different outcomes.

Based on the results from this section we can come to a few conclusions. For example, the privative model defined over univalent primitives and interpreted with CPL logic is extensionally equivalent to the privative model defined over bivalent primitives and interpreted with CPL logic. That is, the set of natural classes that are defined from each model are identical. This suggests that it is the logical interpretation language that is doing most of the heavy lifting when modeling this type of feature system. The same thing can be said about the full feature systems, except it is a CNPL rather than CPL logical interpretation that is the important aspect of representing a full feature system. When it comes to contrastive feature systems, we see that it is in fact the representational aspect that is most important for ensuring that the model theoretic representation is in line with the feature table off of which it is based.

5 Discussion

The previous section showed how the combination of representational primitives and logical interpretation languages results in the ability to describe different types of feature systems. To be complete, we could also consider CNPL with bivalent primitives. Since using negation in the logic forces every feature to be binary, it should be no surprise that it is only the full system that can be correctly represented with this paring of primitives and logic. That being said, this would make it so negative features emerge from both the logic and the primitives which means there is a lot of redundancy built into the system.

So far, the discussion of 0 values has been focused on underspecification, but 0 is used for other things as mentioned earlier. One of the ways in which 0 values are used is for equipollent features such as the place features [labial], [coronal], [dorsal]. If these features are used in a full feature system, then it must be the case that they are interpreted as being binary. Consequentially, `Coronal` and $\neg$`Coronal` must exist as natural classes. It has sometimes been argued that [-coronal] is not a natural class (Yip, 1989). We can take away from this that in order to have any 0 values in a feature system, we cannot use negation in the interpretation language. This goes against most mathematical treatments of phonological features and natural classes (Keenan and Moss, 2009; Ojeda, 2013).

On the other hand, CNPL easily prevents any element from being both $+$ and $-$ for the same feature due to the law of excluded middle. It is logically impossible for any element x to satisfy both $F(x)$ and $\neg F(x)$. If we instead encode the $+$ and $-$ values directly into the primitives, there is nothing about the interpretation language that prevents any element x from satisfying both $+F(x)$ and $-F(x)$.

One option when evaluation $\mathcal{M}^\beta$ with CPL would be to specify feature cooccurrence restrictions. This would be a logical statement with subparts such as $\varphi(x) = \neg[+F(x) \vee -F(x)]$ which would be true only if a segment does not have both the positive and negative features. A model of a feature system $\mathcal{M}$ would therefore only satisfy φ if it did not allow for any segment to be both positive and negative for the same feature.

The goal of this paper was not to find the correct feature system. Rather, the goal was to see how to best represent each of the three different feature systems formally in order to better understand what the differences between each system are. Meaningful differences between the three systems do in fact emerge. For example, privative feature systems can be represented most simply as they minimally require univalent primitives and CPL logic. In order to describe a full feature system there needs to be either an increase in logical power (CNPL) or an increase in representational primitives (bivalent primitives). A contrastive feature system is the least flexible in how it can be represented as it requires CPL and bivalent primitives.

Deciding which of these are the "right" feature system cannot be directly decided based on the analysis provided in this paper. For example, a reviewer points out that most feature systems in use do use a combination of +,-, and 0 which would suggest that CPL with bivalent primitives is on the right path. This raises the question of what it means to be a minus feature in this type of system. If a minus feature is not a negated positive feature (its complement), then why use plusses and minuses at all? Answers to these types of questions lie beyond a purely formal account which is why the analysis given in this paper primarily provides a roadmap for future work on phonological feature systems and a better understanding of how to represent them in formal computational systems.

6 Conclusion

This paper used model theory and logic to explore three types of phonological feature systems commonly used in phonological analysis. The main takeaway is that using negation in the logical interpretation language (e.g., CNPL) forces every feature to be binary. Furthermore, in order to include non-binary oppositions in a feature system, the valuation of the features can be directly encoded into the primitives. One advantage of encoding feature valuation into the primitives is that it allows for the mixture of different types of oppositions without any noticeable issues. This opens the door for more inquiry into how phonological features can and should be viewed in a formal system.

7 Acknowledgments

I would like to thank Jeffrey Heinz, Charles Reiss, Karthik Durvasula, and members of the Stony Brook/Rutgers spring 2021 MathLing reading group for helpful comments and discussion on this material. I would also like to thank the anonymous reviewers for their constructive feedback.

References

Alan Bale and Charles Reiss. 2018. *Phonology: A formal introduction*. MIT Press.

Jane Chandlee, Remi Eyraud, Jeffrey Heinz, Adam Jardine, and Jonathan Rawski. 2019. Learning with partially ordered representations. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 91–101, Toronto, Canada. Association for Computational Linguistics.

Noam Chomsky and Morris Halle. 1968. *The sound pattern of English*. Harper & Row.

Bruno Courcelle. 1994. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126(1):53–75.

San Duanmu. 2016. *A theory of phonological features*. Oxford University Press.

Joost Engelfriet and Hendrik Jan Hoogeboom. 2001. MSO definable string transductions and two-way finite-state transducers. *ACM Transactions on Computational Logic (TOCL)*, 2(2):216–254.

Edward Flemming. 2001. Scalar and categorical phenomena in a unified model of phonetics and phonology. *Phonology*, 18(1):7–44.

John Anton Goldsmith. 1976. *Autosegmental phonology*. Ph.D. thesis, Massachusetts Institute of Technology.

Bruce Hayes. 2011. *Introductory phonology*. John Wiley & Sons.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Sharon Inkelas and Stephanie S Shih. 2016. Re-representing phonology: consequences of q theory. In *Proceedings of NELS*, volume 46, pages 161–174.

Roman Jakobson, C Gunnar Fant, and Morris Halle. 1951. *Preliminaries to speech analysis: The distinctive features and their correlates*. MIT press.

Adam Jardine, Nick Danis, and Luca Iacoponi. 2021. A formal investigation of q-theory in comparison to autosegmental representations. *Linguistic Inquiry*, 52(2):333–358.

C Douglas Johnson. 1972. *Formal Aspects of Phonological Description*. De Gruyter Mouton.

Edward L Keenan and Lawrence S Moss. 2009. *Mathematical structures in language*. CSLI.

Michael Kenstowicz and Charles Kisseberth. 1979. *Generative phonology: Description and theory*. Academic Press.

D Robert Ladd. 2014. *Simultaneous structure in phonology*, volume 28. OUP Oxford.

Leonid Libkin. 2004. *Elements of finite model theory*. Springer.

Connor Mayer and Robert Daland. 2020. A method for projecting features from observed sets of phonological classes. *Linguistic Inquiry*, 51(4):725–763.

John J McCarthy. 1988. Feature geometry and dependency: A review. *Phonetica*, 45(2-4):84–108.

Jeff Mielke. 2008. *The emergence of distinctive features*. Oxford University Press.

Chris Oakden. 2020. Notational equivalence in tonal geometry. *Phonology*, 37(2):257–296.

David Odden. 2005. *Introducing phonology*. Cambridge university press.

Almerindo E Ojeda. 2013. *A Computational Introduction to Linguistics: Describing Language in Plain PROLOG*. CSLI.

Charles Reiss. 2003. Quantification in structural descriptions: Attested and unattested patterns. *The Linguistic Review*, 20:305–338.

Donca Steriade. 1995. Underspecification and markedness. In John Goldsmith, editor, *The Handbook of Phonological Theory*, pages 114–174. Wiley-Blackwell.

Kristina Strother-Garcia. 2019. *Using Model Theory in Phonology: A Novel Characterization of Syllable Structure and Syllabification*. Ph.D. thesis, University of Delaware.

Kristina Strother-Garcia, Jeffrey Heinz, and Hyun Jin Hwangbo. 2016. Using model theory for grammatical inference: a case study from phonology. In *Proceedings of The 13th International Conference on Grammatical Inference*, volume 57 of *JMLR: Workshop and Conference Proceedings*, pages 66–78.

Nikolai Sergeevich Trubetzkoy. 1939. *Principles of phonology*. ERIC.

Mai Ha Vu, Ashkan Zehfroosh, Kristina Strother-Garcia, Michael Sebok, Jeffrey Heinz, and Herbert G. Tanner. 2018. Statistical relational learning with unconventional string models. *Frontiers in Robotics and AI*, 5(76):1–26.

Moira Yip. 1989. Feature geometry and cooccurrence restriction. *Phonology*, 6(2):349–374.