

Subsumption in Feature Theory and Speech Recognition

Author(s): Mortiz Neugebauer

Proceedings of the Twenty-Ninth Annual Meeting of the Berkeley Linguistics Society: General Session and Parasession on Phonetic Sources of Phonological Patterns: Synchronic and Diachronic Explanations (2003) pp. 307-319

Please see “How to cite” in the online sidebar for full citation information.

Please contact BLS regarding any further use of this work. BLS retains copyright for both print and screen forms of the publication. BLS may be contacted via <http://linguistics.berkeley.edu/bls/>.

The Annual Proceedings of the Berkeley Linguistics Society is published online via [eLanguage](#), the Linguistic Society of America's digital publishing platform.

Subsumption in Feature Theory and Speech Recognition

MORITZ NEUGEBAUER
University College Dublin

0. Introduction

Most phonological approaches to the internal structure of speech sounds agree on the general assumption that the layer known as the *segmental layer* is to be interpreted as an abbreviated notational variant of a fully specified feature set, and that this set of features has a complex inherent internal structure. Various graphical formalisms have been applied to visualize dependencies among phonological features ranging from feature tables over tree structures to attribute-value matrices. However, in this paper it will be shown that neither pair of these representations is equivalent from a formal point of view. This paper claims that feature tables as well as feature trees lack a formal definition of *subsumption* establishing a feature hierarchy apart from the graphical notation itself. Therefore, a novel feature theory is proposed, motivating the notion of a subsumption relationship among supersets and subsets of features by (partial) class behaviour. We define how to represent features "if they consistently behave as a unit" (Clements 1985:225).

The paper is organized as follows: section 1 introduces representations which are common in linguistic approaches to structure below the segmental level while simultaneously providing for a critical review regarding the interpretation of the representational means in use. By departing from work on exclusively phonological feature structure, the principal interest of this paper turns to a notion of *feature descriptions* which is intended to establish a logic which allows statements on the decidability and consistency of actual feature terms to be made (cf. section 2). On this basis, section 3 sketches applications of the previously developed feature theory in actual speech recognition systems, focussing on issues of the representation of featural dependencies in the lexicon. The paper concludes with a brief summary and a discussion of topics for future research.

1. Subsegmental Phonological Representations and their Interpretation

An extensive body of research, largely independent from constraint-based or rule-based assumptions, builds on a representational layer below the segment. Recently, the observation that the set of phonological features is not free of but

rather determined by featural interdependencies has attracted remarkable attention. In the following, the representational apparatus of previous theoretical approaches will be revised and it will be subject to significant modifications to capture basic logical relations between the smallest units of phonological description. Beginning with the concept of binary feature specification visualized in feature tables, the forthcoming subsections explore the expressivity of phonological feature representations from a logical point of view. The review proceeds to the characterization of feature trees and finally of attribute-value matrices as applied in numerous prior analyses couched in computational phonological frameworks.

1.1 Phonological Tables

According to assumptions initially formulated in Linear Phonology (cf. Chomsky/Halle 1968), phonological representations consisted of a concatenated bundle of (mostly) articulatory features which are specified in a binary fashion. The internal organisation, i.e. the data structure of these feature bundles in this conception should not to be mistaken for a linked list as it lacks for example the ability to reflect even local precedence relations between individual features. In particular, the linear ordering in visual representations is completely arbitrary and therefore not applicable as a structural reference in the formulation of phonological rules. On the other hand, without resorting to evidence from non-phonological information in a strict sense such as articulatory phonetics, the identical behaviour of two features captured in a phonological rule always seems to occur accidentally. The feature table below gives a feature table representation for a three-member set of commonly used features:

(1)

voiced	±
spread	±
constricted	±

A model of feature organisation of this particular kind makes false predictions whilst claiming that features can be combined without any constraints reflecting the set of possible feature combinations (i.e. sounds) in natural languages and/or constraints establishing classes of features e.g. for access by the rule component (cf. Broe 1992:155). Assuming that the feature set given in (1) is subject to assimilation processes in a certain language, the above representation neither suggests similar phonological behaviour of the features in question, nor does it seem possible to associate just these features – apart from the rest of a remaining feature bundle – with two adjacent skeletal positions. The natural class of laryngeal features, which is generally assumed to consist of exactly the feature set depicted in (1), can only be captured by the invention of an additional feature [\pm laryngeal] along with a set of implication rules of the kind: [$+$ voiced] \rightarrow [$+$ laryngeal]. However, this solution does not provide us with an elegant

foundation for phonological classification since feature classes can only be described by simultaneously inventing new features (such as [±laryngeal]). In addition, these features would lack any segmental equivalent in a phonological system, given that there is no such sound whose place of articulation is sufficiently described by [±laryngeal] since in any case further rules would have to be applied. The subsumption relation encoded can be paraphrased as “every [+voiced] segment is also [+laryngeal] but not vice versa”.

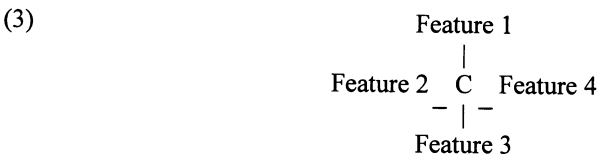
Leaving feature table representations behind, the following section focusses on the contributions of Feature Geometry to the task of phonological classification also referred to as non-linear representations.

1.2 Phonological Trees

The previous subsection drew attention to the observation that feature tables cannot encode local precedence relations as in linked lists for example. However, following the lines of research in Autosegmental Phonology (Archangeli 1985) exactly these relations were captured within a coplanar model of featural organization as given in (2). The representation can be interpreted as a non-branching tree stating that except for feature A, every feature depends on the presence of further features.



The strong claims made in this model appear to be too restrictive for the phonological domain: there is no such phonological feature which is obligatory for every C-element in a given language. The related multiplanar approach, which is based on the idea that any individual feature can behave independently of the rest of the feature set on an individual representational layer faces the same problems as the linear model since no means for featural classification are provided. Table (3) gives a sample representation:



As a model which combines both feature dependencies and independencies, the inspiration for Feature Geometry dates back to *The Sound Pattern of English*

(Chomsky and Halle 1968) where the heterogeneous nature of unordered feature bundles was already noticed. Based on the observation that “it is less apparent but nevertheless true that the feature matrix formalism incorporates certain implications for feature organization that do not follow from the vaguer notion of bundle” Clements (1985:225) elaborates the model of Feature Geometry which introduces class nodes in addition to phonological features. These nodes define feature classes which hence can take part in phonological processes in a homogeneous way. Motivated by the work on the hierarchical organisation of phonological features by Clements (1985), Lahiri and Reetz (2002) apply a tree-based organisation for place of articulation features in their FUL (fully underspecified lexicon) system for speech recognition. The important question with regard to the topic of this paper is in how far class nodes such as *Articulator* and *Aperture* improve a feature system like the one reviewed in the previous subsection where feature classes are captured by the means of additional features.

According to Clements (1985:229) there is a clear-cut distinction between class nodes and features. Whereas the concept of phonological features remains similar to the SPE in the sense that features specify segmental properties, the content of class nodes differs remarkably since “their feature content is entirely determined by the features which they dominate”. This states that class nodes cannot exist on their own but that they rather define class membership relations among sets of phonological features (cf. Bird 1991:137).

A very important aspect that follows from the above definition is that a class node is in contrast to a feature not specifiable but can rather be defined as the set of the contexts of its daughters (cf. Kornai 1994:26). This property can be derived and does not need to be motivated on the basis of empirical investigations, by an explicit activation condition for class nodes (cf. Avery and Rice 1989:183), by some theory-specific reformulation (cf. Padgett 2002) or by the assumption that class nodes are privative features (cf. Wiese 2002). It can be added, that in the latter argumentation the same strategy as in linear phonology is applied since new features are proposed to classify (at least) two others. Another common graphical notation for feature hierarchies besides features trees is the attribute-value matrix which is described in more detail in the following section. Such matrices provide a visualization for *feature descriptions* as they are applied in the forthcoming core part of the paper. Once the feature formalism is developed it will be applied in finite state phonology for the purpose of speech recognition (cf. section 3).

2. Feature Descriptions

In constraint-based grammar formalisms and in logic programming languages various feature descriptions are employed and feature descriptions date back to founding work on Functional Grammar (cf. Kay 1979), though for the rest of this paper we will follow the formalizations in Backofen and Smolka (1995).

The common notational primitives of feature descriptions are functional attributes called features. This notion of *feature* crucially differs from its use in

the previous sections since from now on we will distinguish between *sorts* and *features* while phonological features as used before are sought to correspond most closely to sorts.

The descriptions considered in this paper are obtained from a signature of binary and unary predicates called features and sorts, respectively. Built over the sort alphabet and the feature alphabet, feature descriptions are first-order formulae where in admissible interpretations features must be functional relations and distinct sorts must be disjoint sets. A feature description written in attribute-value matrix (AVM-) notation is the one in (4) a. It is stated that there exists a segment x which carries a place feature, more precisely a height feature LOW whose value is specified as negative. While PLACE and LOW are sorts, *segment* and *height* are encoded as features. It should be noted both that sorts can take complex values (cf. PLACE) and that each bracket in an AVM is indexed with a feature. That index would e.g. define that LOW can take boolean values only. Of course, each matrix can be written in plain first-order syntax as shown in (4) b.

(4) a.

$$\text{segment} \left[\text{PLACE} \underset{\text{height}}{\left[\text{LOW} : - \right]} \right]$$

b.

$$\exists P (\text{segment}(x) \wedge \text{place}(x,P) \wedge \text{height}(P) \wedge \text{low}(x,-))$$

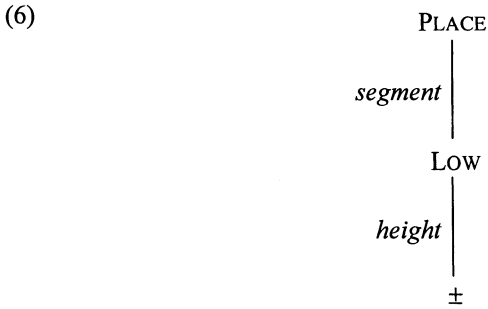
This paper does not present the first attempt to capture phonological feature structures in a strict sorted feature logic. For instance, Bird and Klein (1994) developed a prosodic type hierarchy along similar lines. However, one striking difference between the two approaches seems to be the missing feature indication in the AVM below, where none of the internal brackets comes with further featural information (cf. Neugebauer to appear).

(5)

$$\text{segment} \left[\begin{array}{l} \text{LARYNGEAL :} \\ \\ \\ \text{SUPRALARYNGEAL :} \end{array} \left[\begin{array}{l} \text{SPREAD :} \quad \text{boolean} \\ \text{CONSTRICTED :} \quad \text{boolean} \\ \text{VOICED :} \quad \text{boolean} \\ \\ \text{MANNER :} \quad \left[\begin{array}{l} \text{NASAL :} \quad \text{boolean} \\ \text{CONTINUANT :} \quad \text{boolean} \\ \text{STRIDENT :} \quad \text{boolean} \end{array} \right] \\ \\ \text{PLACE :} \quad \left[\begin{array}{l} \text{CORONAL :} \quad \text{boolean} \\ \text{ANTERIOR :} \quad \text{boolean} \\ \text{DISTRIBUTED :} \quad \text{boolean} \end{array} \right] \end{array} \right] \right]$$

A notational variant of feature descriptions are feature graphs. Edges in feature graphs are labelled with features while nodes are labelled with sorts and thus feature descriptions are interpreted over feature trees in the following way: every sort symbol is taken as a unary predicate where a sort constraint $A(x)$ holds if and

only if the root of the graph x is labelled with A . Contrastively, features symbols are taken as binary predicates where a feature constraint $f(x,y)$ holds if and only if the graph x has the direct subgraph y at feature f . While we can represent the first-order term in (4) b. as a feature graph in a straightforward fashion (cf. (7)), it should be obvious that missing feature labels as in (5) would lead to an ill-formed feature graph which would have to appear on the arcs of a well-formed graph structure. We already stated that the edges of the graph are called features whilst this is of course only observable in complex graphs of feature-value pairs where each value is either again a complex graph or an atomic one. An atomic graph is just a symbol; that is, it contains no features. In figure (6) the leaf of the graph is an atomic value, i.e. “-”. It is this basic aspect of feature graphs which disallows the appearance of specified class nodes as mentioned in section 1.2. where complex graphs such as PLACE in (6) are mistreated as atomic graphs.



In these preliminary remarks on feature graphs the notion of *subgraph* has already been introduced. This aspect as well as the basic axioms of our feature theory are to be defined in the subsequent sections. Since all the following builds on predicate logic, the necessary notions are recalled briefly to conclude this introductory section.

We assume a set SOR of unary predicate symbols called sorts and a set FEA of binary predicate symbols called features. Given a feature structure, a sequence of labels is used to extract a substructure. Such a sequence of features is called a path over the set of all features. The symbol ε denotes the empty path which means that for every path p there exists a set of equivalent paths $\varepsilon p = p = p\varepsilon$. In the first case, ε is a prefix of p meaning that there exists a path p' such that $\varepsilon p' = p$. In addition to the partially disjoint sets SOR and FEA we assume an equally disjoint set of variables. Under our signature $\text{SOR} \cup \text{FEA}$, every term is a variable and an atomic formula is either a feature constraint $f(x,y)$, a sort constraint $A(x)$, an equation $x = y$, \top or \perp . In all these cases x, y denote variables, A denotes a sort and f a feature. We speak of a theory as a set of closed formulae where a model of a theory is a structure that satisfies every formula of a theory. The following section defines a theory by means of three axiom schemes. These axiom schemes are inspired by work the of Backofen and Smolka (1995).

2.1. Axioms

As stated at the beginning of section 2, it is assumed that features are functional which can be demonstrated given two instances $f(x,y)$ and $f(x,z)$ of a feature constraint f , where x, y, z are variables. If both constraints are supposed to be elements of an identical formula ϕ then y and z must be identical:

$$(7) \quad \forall x \forall y \forall z (f(x,y) \wedge f(x,z) \rightarrow y = z) \text{ for every feature } f$$

Sorts are required to be mutually disjoint sets. Thus the second axiom scheme says that if a sort constraint A holds for a variable x in a formula ϕ and there is a sort constraint B over x defined in the same formula, then A and B must be identical. We capture this by stating the second axiom as in (8) which defines that the statement that a variable x is as well of sort A and B is false.

$$(8) \quad \forall x (A(x) \wedge B(x) \rightarrow \perp)$$

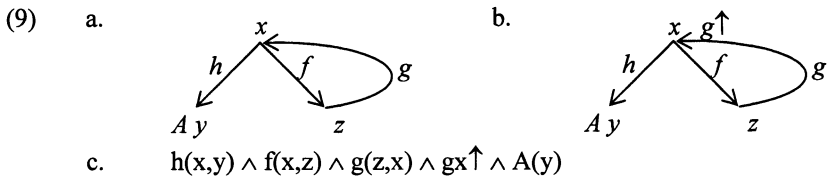
Thus far both axiom schemes are formulated on the general assumption that all features and sorts are non-empty and that feature descriptions are consistent and have solutions. To actually guarantee these properties of our feature theory a third axiom scheme is required stating that consistent feature descriptions are satisfiable. For this reason we introduce the notion of a solved clause to describe the satisfiability of feature descriptions. A solved clause can be seen as the graph whose nodes are the variables appearing in the clause and whose arcs are given by the feature constraints. Nodes are labelled by sort constraints or by exclusion constraints which state a feature f is undefined on a variable x . We will represent exclusion constraints as $x\uparrow$ saying that f is *undefined on* x . Formally a solved clause is a conjunction σ of atomic formulae i.e. feature and sort constraints as well as exclusion constraints. The latter kind, the exclusion constraints, are equivalent to $\neg \exists y f(x,y)$ for some variable $y \neq x$. A variable x is constrained in a solved clause σ if σ contains one of these constraints on x . Based on these notions we can finally state the third axiom scheme: $\exists X \sigma$ (for every solved clause σ and $X = CV(\sigma)$).

The feature theory developed thus far is the set of all sentences that can be obtained as instances of the axiom schemes in (8 – 10). According to these three axiom schemes, a solved clause σ must satisfy the following four conditions. The most obvious condition is that no atomic formula such as an equation, sort or feature or an exclusion constraint is supposed to occur twice in σ . In addition, sorts must be mutually disjoint sets, features are supposed to be functional and finally if a feature is defined on a specific variable then it cannot be the case that there is an exclusion constraint on this specific variable and this specific feature in an identical solved clause σ . An example of a graph representation of a solved clause will be given in the following section since it focusses on models of our feature theory in terms of feature graphs.

2.2. Feature Graphs

Building on the axioms defined above, it is the aim of this section to provide for a model of our feature theory which allows us to represent feature structures – here – in the area of phonology. The graph representations which show a certain degree of similarity to the tree diagrams commonly used in Feature Geometry. Thus, we introduce a representational formalism which provides us with illustrative means for application in section 3.

As we introduced our three pairwise disjoint sets of variables, sorts and features it was noted that we can extract a substructure of a feature description using a sequence of labels. Such a sequence of labels is called *path* and is defined as an expression over FEA^* , the set of features. Before we proceed to give a graph representation of a solved clause, we introduce the notion of *feature pregraph* which is a pair of a solved clause σ as defined in the previous section and a variable x . In the case of feature pregraphs, the solved clause does not contain exclusion constraints such that for every variable y occurring in σ , there exists a path p from x to y in the solved clause σ (i.e. $p(x,y) \in \sigma$). A feature pregraph is called *subpregraph* of a feature pregraph if the solved clause of the pregraph subsumes the solved clause of the subpregraph and the two variables in question are identical or x is an element of the set of variables defined in the subpregraph. Note that feature graphs are directed graphs with a distinguished root node.



In the example (9) a. the variable y carries the sort symbol A whereas the nodes labelled with x and z are not specified in a similar way. This is possible only in feature graphs whilst in feature trees each node must carry a sort symbol (cf. Backofen and Smolka 1995). In addition to the pregraph in (9) a., a feature graph contains the exclusion constraints as mentioned above. The above table contains the graph representation in (9) b. as well as the corresponding solved clause in (9) c.

3. Applications in Speech Recognition

This last section argues that subsumption relations between finite state automata modelling phonotactic regularities motivate a set of feature graphs for lexical knowledge representation in terms of supergraphs and subgraphs, respectively. Since we aim to define transitions over sets of sounds rather than atoms we make reference to a specific line of research in the field of finite state methods in natural language processing, known as *predicate augmented finite state automata*

(in the following *pfsa*; cf. van Noord/Gerdemann 2001). This will be sketched in section 3.1. Our novel approach to phonological feature descriptions in combination with the previously mentioned augmented finite state techniques will build on the model of *Time Map Phonology* (cf. Carson-Berndsen 1998), a constraint-based model of speech recognition. Since this model implements restrictions on syllable well-formedness in terms of finite state techniques by defining phonotactic automata, we present a homogenous model in section 3. We show that we can finally derive complex feature graphs from a regular grammar for a selective data set of laryngeal features in German. Consequently, language-dependent phonological feature classes are presented as a result. These feature classes are regarded as a significant contribution to lexical knowledge in computational recognition systems since they allow to further determine the phonotactic context. We agree with Fant (1970:210) that feature classes constitute an essential component of the phonological lexicon.

3.1. Integrating Feature Descriptions and Finite state models

Most recently, some approaches have proposed finite state automata where transitions are associated with sets of symbols (cf. Bird and Ellison 1994, Walther 1999). Partially motivated by these developments van Noord/Gerdemann (2001) present finite state machines in which atomic symbols on the transition arcs are replaced by arbitrary unary predicates. In our feature theory unary predicates are used for a very similar purpose, namely as sort constraints labelling the nodes of feature graphs. Below we briefly describe the concept of predicate augmented finite state automata (called *recognizers* by van Noord/Gerdemann 2001) and consequently replace their set of arbitrary predicates with a set of sort constraints.

In contrast to a standard finite state automaton, a *pfsa* A is specified by a six-tuple $(\Delta, \Sigma, \Pi, E, S, F)$ just as described below:

- (10)
- the finite set of states Δ
 - the input alphabet Σ
 - Π the set of predicates over Σ
 - E a finite set of transitions $\Delta \times (\Pi \cup \{E\}) \times \Delta$
 - the set of start states $S \mid S \subseteq \Delta$ and
 - the set of final states $F \mid F \subseteq \Delta$

We can extend E to a function E^\wedge and define the relation $E^\wedge \subseteq \Delta \times \Sigma^* \times Q$ by induction. For this purpose we state in a first step that for all states q in Δ there exists an ε -transition which results in the same state q . Similarly, for all transitions (p, ε, q) in E^\wedge there exists a corresponding transition (p, ε, q) in E . A *pfsa* is called ε -free if there are no $(p, \varepsilon, q) \in E$. For all other transitions (q_0, π, q) in E and for all $\sigma \in \Sigma$ if $\sigma(\pi)$ there exists a corresponding transition (q_0, π, q) in E^\wedge . As a last step, if (q_0, x_1, q_1) and (q_1, x_1, q_1) are both in E^\wedge then we can infer that (q_0, x_1, q_1) is also an element of E^\wedge .

Finally, the language accepted by A is defined to be $\{\omega \in \Sigma^* \mid q_S \in S, q_F \in F, (q_S, \omega, q_F) \in E^\wedge\}$ (cf. van Noord/Gerdemann 2001:266).

Our extension now consists in replacing Π , the set of predicate symbols with a set of sort constraints which comes equipped with all properties defined in section 2.1. One significant property concerns the determinization of automata where van Noord/Gerdemann mention on the one hand well-known aspects like the requirement of just single start state and like the existence of at most one transition (q, π, q') for all states and symbols such that $\sigma(\pi)$. On the other hand they have to exclude overlapping transitions, i.e. intersecting sets of segments. To exclude these cases where more than one transition is applicable one has to compute appropriate individual transitions for all subsets. Thus for two such individual overlapping transitions over sets A and B in a *pfsa* we finally end up with three transitions for $A \wedge B$, $\neg A \wedge B$ and $A \wedge \neg B$. In our approach, this computation can hugely benefit from our inherently structured graph representations and the related possibility to extract subgraphs. Simply by reference to (possibly negated) sets we provide for the predicates which serve as labels for each transition in the automaton. Inspired by van Noord/ Gerdemann, we compute the transitions over a set SOR of sort constraints leaving a given subset Δ . In a first step, we compute the function $Trans^\Delta : SOR \rightarrow 2^Q$, defined as $Trans^\Delta(\pi) = \{q \in Q \mid p \in \Delta, (p, \pi, q)\} \in E$. Thus for instance, suppose $\Delta = \{q\}$ along with the transitions

$$(11) \quad E = \{ (p, \pi_1, q_1), (p, \pi_1, q_2), (p, \pi_2, q_2), (p, \pi_2, q_3), (p, \pi_2, q_4) \}$$

Consequently, the logical combinations of sets of states for each of the sort constraints π_1, π_2 are the following:

$$(12) \quad Trans^\Delta(\pi_1) = \{q_1, q_2\}, \quad Trans^\Delta(\pi_2) = \{q_2, q_3, q_4\}$$

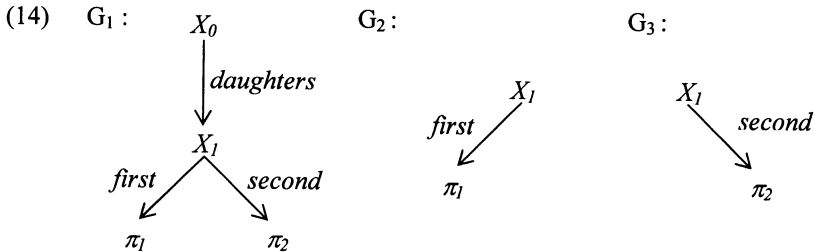
We assume that SOR' is the set of sort constraints in the domain of $Trans^\Delta$ and receive the following transitions over sorts as a result. Since we are dealing with two sort constraints as predicates and a negation operator, four possible combinations have to be accounted for.

$$(13) \quad \begin{aligned} & (\Delta, \pi_1 \wedge \pi_2, \{q_1, q_2, q_3, q_4\}) \\ & (\Delta, \neg \pi_1 \wedge \pi_2, \{q_2, q_3, q_4\}) \\ & (\Delta, \pi_1 \wedge \neg \pi_2, \{q_1, q_2\}) \\ & (\Delta, \neg \pi_1 \wedge \neg \pi_2, \{\emptyset\}) \end{aligned}$$

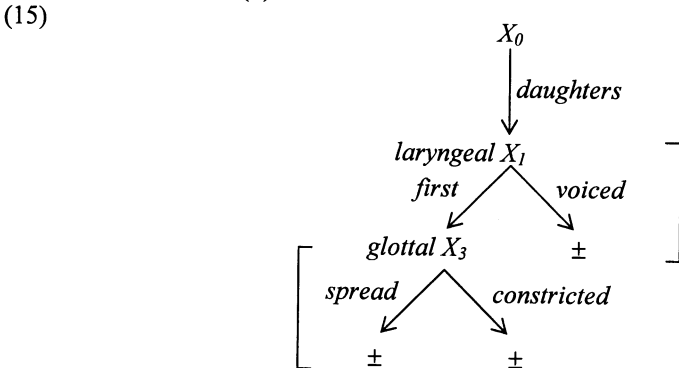
The contribution of feature descriptions and corresponding feature graphs can be demonstrated by the three related graphs in (14). Graph G_1 is the most specific one and therefore represents the simple conjunction of the sort constraints π_1, π_2 . G_1 subsumes G_2 and G_3 while the latter two graphs are not connected in any

Subsumption in Feature Theory and Speech Recognition

subsumption relation. The negation of both constraints naturally does not meet any graph representation, at least not for the given predicates π_1, π_2 .



To exemplify the predicate augmented finite state approach as described above we give the following complex laryngeal tree including the three phonological features introduced in (1).



For a precise application of this feature graph structure for determinization purposes of finite state machines we refer to a computation along the lines of (11 - 13).

4. Summary and Concluding Remarks

Based on a review of representational means commonly used in phonological feature theory, this paper presented a novel approach in terms feature descriptions for the internal structure of speech sounds. For this purpose we concentrated on feature theory in constraint-based grammar formalisms and in logic programming languages to define a theory of feature descriptions and a corresponding model of feature graphs.

Equipped with these foundations we integrated our representations with finite state techniques and concluded with an exemplification in the field of speech technology.

Since the paper primarily focussed on logical refinements rather than the analysis of speech data the latter has to be a major topic in future research. Another desirable issue is to explore our approach for finite state transducers rather than automata since various applications in speech recognition rely on these machines (cf. Carson-Berndsen 1998). Special attention has to be given to determinization and minimization of transducers as pointed out in van Noord/Gerdemann (2001).

5. Acknowledgements

This material is based upon works supported by the Science Foundation Ireland under Grant No. 02/IN1/I100. The opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Science Foundation Ireland.

References

- Archangeli, D. 1985. Yokuts harmony: Evidence for coplanar representation in nonlinear phonology. *Linguistic Inquiry* 16: 335-372.
- Avery, P. & K. Rice. 1989. Segment structure and coronal underspecification. *Phonology* 6: 179-200.
- Backofen, R. & G. Smolka. 1995. A complete and recursive Feature Theory. *Theoretical Computer Science* 146: 243-268.
- Bird, S. & E. Klein. 1994. Phonological Analysis in Typed Feature Systems. *Computational Linguistics* 20: 455-91.
- Bird, S. & M. E. Ellison. 1994. One-Level Phonology: autosegmental representations and rules as finite automata. *Computational Linguistics* 20: 55-90.
- Bird, S. 1991. Feature structures and indices. *Phonology* 8: 137-144.
- Broe, M. 1992. An introduction to feature geometry. In G. J. Docherty & D. Ladd (eds.) *Papers in Laboratory Phonology*, 149-165. Cambridge: Cambridge University Press.
- Carson-Berndsen, J. 1998. *Time Map Phonology: Finite State Models and Event Logics in Speech Recognition*. Dordrecht, Holland: Kluwer.
- Chomsky, N. & M. Halle. 1968. *The Sound Pattern of English*. New York: Harper and Row.
- Clements, G. N. 1985. The Geometry of Phonological Features. *Phonology Yearbook* 2: 225-252.
- Fant, G. 1970. Automatic Recognition and Speech Research. Reprinted in G. Fant 1973. *Speech Sounds and Features*, 210-215. Cambridge, MA: MIT Press.
- Kay, M. 1979. Functional Grammar. In *Proceedings of Fifth Annual Meeting of the Berkeley Linguistics Society*. Berkeley, CA : UC Berkeley.
- Kornai, A. 1994. *Formal Phonology*. Garland Publishing.

Subsumption in Feature Theory and Speech Recognition

- Lahiri, A. & H. Reetz. 2002. Underspecified recognition. In C. Gussenhoven & N. Werner (eds.), *Laboratory Phonology VII*, 637-675. Berlin: Mouton.
- Neugebauer, M. to appear. Computational Phonology and Typed Feature Structures. In *Proceedings of CamLing*. Cambridge: University of Cambridge.
- Padgett, J. 2002. The unabridged feature classes in Phonology. *Language* 78.1: 81-110.
- van Noord, G. & D. Gerdemann. 2001. Finite State Transducers with Predicates and Identities. *Grammars* 4 (3): 263-286.
- Walther, M. 1999. One-Level Prosodic Morphology. *Marburger Arbeiten zur Linguistik* 1. University of Marburg.
- Wiese, R. 2002. *The Phonology of German*. Oxford: Oxford University Press.

Moritz Neugebauer
Department of Computer Science
University College Dublin
Belfield, Dublin 4
Republic of Ireland

moritz.neugebauer@ucd.ie