Preferences as defaults in computational phonology

Dafydd Gibbon

18 March 1999 (Version: 11 February, 2001)

Preferences and defaults - typical or untypical?

A *preference* is intuitively an element of a set which is valued more highly than other elements; the preferred element can be interpreted as an object or a property, a state, an event or a course of action. Similarly, a *default* is a *typical state* or *typical choice* which is valued more highly than other choices, other things being equal (other meanings of "default" are of no concern here).

Neither a preference nor a default is a fixed value: a preference may not be favoured under all conditions, and a default may be overridden by other considerations in specific situations. The concept of once-for-all fixed rules is gradually being replaced in linguistics by a notion of preference, starting perhaps with Trubetzkoy's notion of *privative opposition*, though various notions of *markedness* to Vennemann's *Preference Laws* and *Optimality Theory*, as well as the *generalisation and irregularity hierarchies* of computational lexicon theory. Quantitative expressions of markedness range from Labov's *variable rules* to *probabilistic automata* as used in corpus tagging and in automatic speech recognition. The mechanisms used to express preference valuations range from marked/unmarked values in feature structures through default inheritance hierarchies with orthogonal and prioritised inheritance, default automata, to fully fledged default logics and non-monotonic reasoning.

Perhaps the simplest gloss for *default* is *typical*, a word which deserves more extensive empirical and phenomenological analysis than it can be given here. In everyday life, it is a vehicle for stereotypes and many forms of prejudice, the Occam's razor wielded in the daily struggle to avoid as far as possible the chore of thinking rationally. In this pragmatic context, the allies of *typical* are adverbs of subjective frequency, such as *usually*, *mostly*, *probably*, which are used not as quantifiable statements of frequency and probability, but as adverbs of degree which qualify stereotypic convictions.

The main goal of the present study is to not to argue for the use of default mechanisms for explicating preferences in phonology and prosody, since this is quite widely accepted. Rather, I will try to collect a wide range of different aspects of phonology and prosody, and show, sometimes in novel ways, the relevance of defaults to them, with the longer term goal of developing a unified theory of defaults and preferences in phonology and prosody. I will look at the simple example of English plurals to start with. Then I will move to default-override relations in phonetics, prosody, and the lexicon, followed by a detailed discussion of phonotactic (sequential) defaults, referring to a number of types of default automaton. This will be followed by a discussion of linguistic approaches to preferences, including Vennemann's Preference Law theory, and Optimality Theory. Finally I will take a brief look at the Integrated Lexicon approach to modelling default-override relations in the lexicon. It will be a difficult tight-rope act along the line which joins descriptive and computational linguistics, and I ask the reader's indulgence if the presentation wobbles a little too much for comfort to the one or the other side.

Preferences, defaults, ignorance and bliss

In a sense, preferences were invented to help us to avoid complicated decisions, and defaults to help us to live in ignorance. Sometimes ignorance is bliss; defaults define prejudices in everyday life which help to avoid and generate misunderstanding and strife – though as often as not, prejudices in everyday life unfortunately often turn out to be indefeasible. Since ignorance is perhaps the most natural and the most widespread of the mental states, defaults may arguably be taken as an essential component of natural reasoning in general, and of natural linguistic theorising in particular. In fact, defaults contribute towards elegant explanations for a wide range of linguistic facts, and are "typically" concerned with the following kinds of reasoning in theoretical phonology:

1. Generalisations about "normal properties" of phonological objects (markedness of features specifications, i.e. roughly: weighting of attribute values).
2. Orderings over sets of generalisations with respect to their specificity (rule ordering by specificity of context, and some uses of the "elsewhere condition" .
3. Rule application principles in the absence of information about other contexts, overridable by interaction with other principles (one interpretation of Optimality Theory).
4. Heuristic abbreviations over a finite vocabulary (some uses of the "elsewhere condition" .

Defaults, non-monotonic reasoning, and IF-THEN-ELSE hierarchies

A simple example of non-monotonic ordering over sets of generalisations is the English plural affixation rule. This rule is conveniently exemplified by predicates referring to farmyard animals (*cats, dogs, horses, oxen, sheep* - /kæts dɒgz hɔsɪz ɒksən ʃip/), and is traditionally formulated as a set of ordered morphophonological rules, with shared rule parts indicated by braces; one rule style (without going into details, and using features to abbreviate natural classes of phonemes) is:

$$\{\text{N-PLURAL}\} \rightarrow \left\{ \begin{array}{lll} \text{ɪz} \;/ & [\text{sibilant}] & \_\_\_\_\_\_ \\ \text{z} \;/ & [\text{voiced}] & \_\_\_\_\_\_ \\ \text{s} \;/ & [\text{voiceless}] & \_\_\_\_\_\_ \end{array} \right.$$

Why is this rule set *non-monotonic*? A rule set is *monotonic* if adding a rule leads, as one might expect, to an increase in the number of outputs for a given set of inputs. On the other hand, a rule set is *non-monotonic* (as with non-monotonic functions in algebra) if adding more rules produces fewer outputs for a given set of inputs. If the suffix /-z/ could be added freely to all English nouns, this would be a *monotonic* rule system: add a rule, and the number of possible output forms increases. But in English, this is not really the case, for instance, if the noun ends in a voiceless consonant: add this rule, and the number of possible combinations decreases. And this is not all, because even when the noun ends in a voiced consonant, if this consonant happens to be a sibilant the rule does not apply either: so add another rule, and the number of possible combinations decreases again.

Avoiding discipline-specific notation, a non-monotonic rule set is easily formulated in a familiar IF-THEN-ELSE form (cf. also Kiparsky's *elsewhere condition* in phonological theory):

IF an English word is a lexical exception
THEN its plural must be looked up in the lexicon (e.g. *ox-oxen*; *sheep-sheep*)
ELSE it takes a suffix AND
IF it ends in a sibilant
THEN the suffix is /ɪz/ (e.g. *horse-horses*)
ELSE IF it ends in a voiceless segment
THEN the suffix is /s/ (e.g. *cat-cats*)
ELSE the suffix is /z/ (e.g. *dog-dogs*)

An alternative but equivalent formulation is in terms of defaults and overrides with "UNLESS" , which means "EXCEPT IF" or "IF NOT" , and uses the inverse ordering:

An English word takes the plural ending /z/
UNLESS it ends in a voiceless segment,
in which case it ends in /s/
UNLESS it ends in a sibilant,
in which case it ends in /ɪz/
UNLESS it is a lexical exception,
in which case it must be looked up in the lexicon.

In the present example, the most general default case only covers one context, namely final voiced segments. In fact, in this example the voiceless/voicing conditions could be reversed, indicating some arbitrariness in the decision and a need for independent motivation. The independent motivation could lie in the fact that the voiceless condition applies to far fewer segments, namely only the voiceless obstruents (except /tʃ s ʃ/, i.e. /p t k f θ/), than the voicing condition, while the voicing condition applies not only to the voiced obstruents (except /dʒ z ʒ/) but also to the nasals /m n ŋ/, the glides /j w/ and all the vowels. A hierarchy of constraints like this indicates that the notion of default is itself relative: there is not just one elsewhere condition, but a series of more and more general elsewhere conditions. The ranked constraints of Optimality Theory are related in similar fashion. However, we are not concerned here with discussing linguistic motivation for a particular style of default ordering among constraints, or for a particular analysis of the treatment of plural in English, but with illustrating the default ordering principle itself.

Defaults in the most general sense of the term can simply be a waste bin to cover up for ignorance of the facts, as already noted. Defaults and overrides in this open-ended *semantic* or *pragmatic* sense are widely considered too complex to handle in a formal system. But in describing linguistic forms, as in the present case, the situation is different: the defaults cover a *closed world* which is completely known, and are not used as ignorance flags but to express interesting generalisations about irregularities or markedness inside this closed world which could not otherwise be expressed neatly. Defaults and overrides in a closed world can be spelled out exhaustively in completely monotonic fashion by conjoining positive and negative constraints, the length of the conjuncts being a function of the most specific condition in the default hierarchy. The exhaustive formulation means that IFF (*if and only if*) can be used instead of IF, and that the rules can be considered in any order:

1. The plural of an English word ends in /z/ IFF it does NOT end in a sibilant AND it does NOT end in a voiceless segment AND it is NOT idiosyncratic.
2. The plural of an English word ends in /s/ IFF it ends in a voiceless segment AND it does NOT end in a sibilant AND it is NOT idiosyncratic.
3. The plural of an English word ends in /ɪz/ IFF it ends in a sibilant AND it is NOT idiosyncratic.
4. The plural of an English word must be looked up in the lexicon IFF it is idiosyncratic.

The terms *monotonic* and *nonmonotonic* are used to refer to the fully explicit and the default-based types of constraint. Think of the four cases listed here as axioms from which, given an English word stem, the full plural form can be inferred. Any of these four cases describes a subdomain of English plural forms independently of the others, and collecting any of them simply *monotonically increases* the overall domain described. The UNLESS and IF-THEN-ELSE axiom sets are different. The first axiom purports to describe all English plurals, but adding more axioms *non-monotonically decreases* the overall domain described by this first rule.

The domain of speech sounds

Phonology: irregularity defaults

In a well-known paper, Morris Halle and Sylvain Schlumberger discussed “why phonology is different” . Without going into the details of their discussion, two obvious answers lie in blocking (defaults and overrides) in time-based features of the phonological domain:

1. *History*: language, and in particular the sound pattern of a language, develops on the one hand by grammaticalisation and lexicalisation, and complementarily by accretion and productivity, with conflicts between internal (system inherent) and external (social) constraints leading to inconsistencies, i.e. irregularities.
   Example: prestige pronunciations from other dialects, or loan words, which may introduce extrinsic sound patterns.
2. *Performance*: sound patterns are realised fairly directly by physical systems with partially independent components which are only loosely synchronised with each other and are arranged in physical sequences in which the effects of one component may overshadow the effects of another.
   Example: *in Bielefeld* may be pronounced in English or German as /ɪmbiləfɛlt/, with the alveolar /n/ overshadowed (covered, blocked) by the bilabial /m/ (at later stages of grammaticalisation, not being articulated at all).

But in recent research into the syntax of speech, driven by spoken language engineering requirements, and by research into the formalisation of prosodic phonologies by means of event logics, shows that phonology is not all that different, for instance from syntax, or indeed semantics. Part of semantics is also concerned with events and temporal relations like the phonetic and phonological events and temporal relations into which phonological units are mapped; part of syntax is different from the way in which it is traditionally conceived in logic and linguistics, in being essentially dependent on processing and therefore temporal constraints, as suggested in work by John Hawkins (1994) on performance theories of order and constituency. So it may be that phonology is no different after all, at least not on these grounds.

Phonetics: mapping defaults

Taking current wisdom about phonetic interpretation at face value, I will explicate phonetics as a formal *model* for phonological representations, $INT_{phon} = \langle F_{phon}, P \rangle$, where F_{phon} is the phonetic interpretation (phonetic correlate) function from phonological structures into a representation P of a phonetic domain. The situation is complicated by the fact that P is itself highly complex, and may be analysed as a vector of subdomain representations ordered in a causal-temporal chain: $\langle P_1, ..., P_n \rangle$. Empirically, the phonetic subdomains are characterised by universal physical and temporal constraints. The model is visualised in Figure 1.

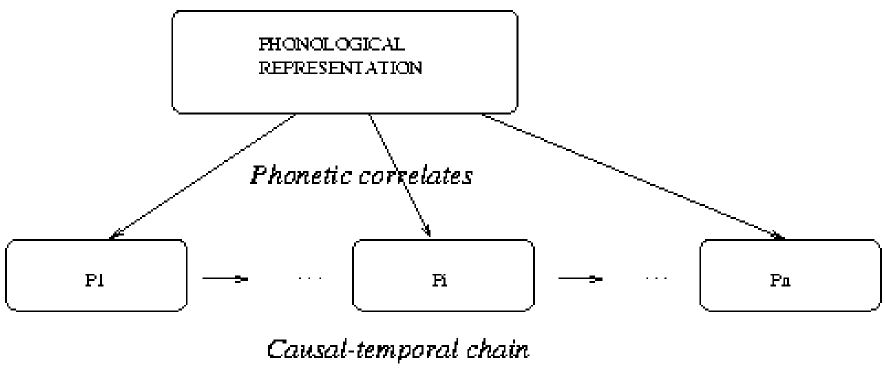


Figure 1: Phonetic domain models.

Depending on the granularity of analysis, if *n*=1 in the model, then we are probably talking about linguistic phonetics, and using an alphabet such as the IPA. In signal phonetics, it is often the case that *n*=3, and we identify a causal-temporal chain from *articulatory phonetics* (sound production) through *acoustic phonetics* (sound transmission) to *auditory phonetics* (sound perception). This is still the folklore of introductions to phonetics. In phonetic research, a much higher granularity with a cyclical chain such as the following (*n*=8) is not uncommon:

1. Central nervous system
2. Efferent nerves
3. Muscles
4. Resonator shape
5. Acoustic channel
6. Inner ear shape
7. Sensors
8. Afferent nerves
9. Central nervous system (= 1.)

This model goes a long way towards explaining why, at least, *phonetics* is different (and thus, indirectly, phonology): because phonetics describes *physical systems* in which *force*, *mass*, *causality* and *time* constraints need to be taken into account.

A monotonic description of the domains of phonetics is logically possible when all stages are taken into account, and carefully distinguished. But in descriptive practice the transformations lead to complex relations between the phonological representations which relate most directly to P_1 and the phonological representations which are directly reconstructable from P_n,. Consequently, taking only phonological representations into account, over the whole causal chain these hidden phonetic complexities look suspiciously like additions and deletions of feature values, and thus like a non-monotonic system.

So do we have a case of *don't know* or *incomplete knowledge* defaults here, which may be superseded by more exhaustive empirical modelling? The answer is surely yes, though not a *knowledge gap* in terms of empirical facts. The account is incomplete because it lacks a semantics for temporal constraints, and in particular which models mappings in terms of *overrides of simultaneous events in the causal chain*. Overrides in this chain include the completion of inputs by top-down information (cf. Marslen-Wilson's *cohort theory* of speech perception), radical (default) underspecification, and issues such as overriding by masking of one articulator by another is Keating (1988).

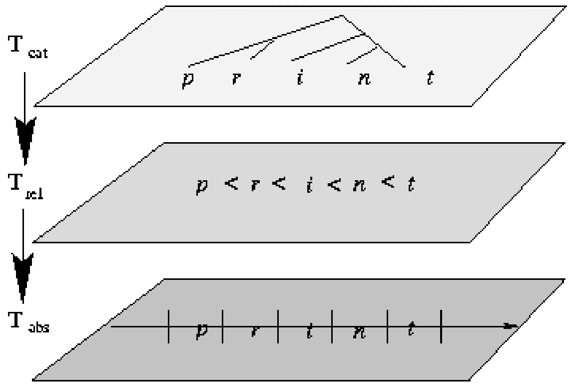
An integrative approach to domain levels: Time Map Phonology

The first attempt at formally integrating temporal constraints into phonology was made by Bird & Klein's *Event Phonology* (1989), designed as an explication of autosegmental phonology. Event Phonology provides a solid starting point for explicating a complex model such as the one outlined above, but does not go far enough. The approach has the advantage of being purely declarative and logically well founded, but falls short in the following respects:

1. The approach is purely syntactic: the purely relational event concept needs to be mapped into a temporal structure with "real time" absolute length constraints.
2. Over and above the procedural semantics given, based on *modus ponens*, an operational semantics is required in which precedence and overlap relations are translated into time-annotated processes in a way which matches human performance: that is, it matters that an utterance may take 3 seconds, but not 3 hours, and that production and perception are incremental, and not instantaneous.
3. Incremental processing implies partial knowledge at any given time with respect to the possible analysis, and a concept of *incomplete event* is required.
   Example: A sentence like *My mother is going home in a while.* is voiced all the way through; a model which requires completion of the full voicing event, i.e. in this case the whole sentence, is clearly

unrelated to a plausible performance theory.

4. Phonological structures are dealt with in isolation from other levels of linguistic description. The assumption of autonomy may or may not be justified, and is also a feature of the autosegmental *explicanda*. Concepts of *lexicon*, of *morphophonology*, of *hierarchical prosodic domains* are ultimately required.
5. The approach is based empirically on the traditional “interesting fragment of the language” paradigm of formal linguistics; a more modern requirement is for a model to be tested on a “representative subset” of the language, in particular on an annotated corpus.



[G2: C:\Eigene Dateien\cpdefaults\IMG3]

Figure 2: Declarative outline of time map phonology mappings.

In a recent study (1998) which investigates the relevance of phonology and phonetics to the applied area of speech recognition, Carson-Berndsen takes a closely argued path through event logics, and finite state automata as models for these logics, and suggests *inter alia* that phonology is characterised by *events in relative time* and phonetics by *events in absolute time*. The terms “rubber time” and “clock time” for these time domains has been used by Andras Kornai. The mapping between abstract categories, relative time categories and absolute time categories is termed *Time Map Phonology* (TMP) by Carson-Berndsen. In Carson-Berndsen’s model, underlying abstract phonological lexical representations are mapped to phonetic models with absolute time annotations (see Figure 2), together with a multi-level operational semantics in the form of an incremental virtual machine concept for event structures at different levels. The basic idea, first put forward by Gibbon (1992b), is that three basic time types need to be distinguished in an adequate theory of phonology:

1. Category time: the temporal “null annotation” , i.e. conventional categories.
2. Relative time (“rubber time”): categories linked by relations of temporal precedence and overlap.
3. Absolute time (“clock time”): relations annotated by absolute “real time” interval lengths.

Each level provides an increasingly realistic denotational semantics, i.e. an interpretation, for the preceding one. If the models are finite state transducers at each level, then all levels in the cascade can be composed into a single large transducer; see Kaplan & Kay (1994). A related model oriented view of phonetic interpretation has been taken by Coleman (1998) and Local in speech synthesis, in which abstract phonological structures are mapped into streams of Klatt synthesiser parameters. Orthographic interpretation and the interpretation of other gestures are analogous models. The relevance of this to our topic is defined via the notion of causal-temporal chain, together with the following postulates:

1. All physical systems can be modelled at some level by finite state devices.
2. Event logics lend themselves to modelling by finite state devices.

In Gibbon (1992b) I suggested that this approach to phonetic interpretation has very much in common, formally, with linguistic semantic interpretation, in being, formally, itself a variety of model based semantic interpretation. In the TMP approach, the level of Category Time is just conventional syntax (here word syntax); the level of Relative Time is analogous to *logical form*, and the level of Absolute Time is analogous to a *domain model*. Phonetic interpretation is impoverished in relation to full linguistic semantic interpretation, in being restricted to time domains, but this opens up a number of new perspectives on the relation between phonetics and semantics:

1. Controversially formulated: phonetics is a subset of semantics; in particular, the phonetic domain is a subset of the semantic domain, and events in the phonetic domain are of the same type as events in the linguistic semantic domain. Why? -
2. Trivially: We can talk about phonetic events ...
3. Less obviously: We can use phonetic events to *refer to* other phonetic events, for instance we can use *pitch accent* to point to the locations of particular constituents of utterances in time (and, via our location as utterer, in space), as a basic focussing mechanism. This functionality in speech I have referred to (in Gibbon 1976 and elsewhere) as the *metalocutionary hypothesis*. Example (due to Bolinger; the target of metalocutionary pointing is the prefix or the syllable): *This whisky was not EXported, it was DEported.*
4. Entirely unobviously: logical oddities like homological and heterological words (‘s*hort*” *is short*, “*Long*” *is short.*), or token reflexives (*This sentence contains five words.*, or perhaps *This inscription is printed in black ink.*) take on less of an odd appearance when the phonetically (or orthographically) interpreted token is located in the same domain as its denotation.

The essential point of the time mapping is the ease with which apparent default-override relations can be explained as declarative mappings between domains: assimilations and reductions are simply not in the same physical domain as lexical information.

Prosody

There are two other strange, and rather different, components in linguistic descriptions, and in general no-one knows exactly how to relate them to the ‘rest of the world’: *prosody*, and the *lexicon*. First of all, prosody.

Prosody is sometimes taken to be somehow different from phonology; part of the problem is terminological. Let me define prosody as “the phonetic interpretation of words and of structures larger than the word” . This definition includes conventional phonology, and a stroke of the terminological pen elimininates the fundamental divide between phonology and prosody. The competing conventional definitions are full of hazards, though easy to understand at a more fuzzy, intuitive level:

- Prosodic categories are those which are larger than the phoneme (often associated with the term ‘suprasegmental”).
- Prosodic categories are those which correlate with pitch, amplitude and duration patterns (often associated with the terms “intonation” , “accent” , “rhythm”).

It is fairly well established that prosody can best be modelled by a hierarchy (perhaps more than one) of prosodic domains: phoneme-sized units are at the bottom, and at the top of the hierarchy is the utterance (whatever that is). The most well-known varieties of prosodic hierarchy are the tagmemic hierarchy introduced by Pike in the forties and fifties, the metrical hierarchy of Liberman & Prince (1977), in the 1970s, and the prosodic hierarchy of Selkirk (1984) and others, in the 1980s. There is a complex mapping between the prosodic hierarchy and lexico-syntactic hierarchies which was first systematised by Halliday in several studies in the late 1960s. There is also a complex mapping between the prosodic hierarchy and phonetic features; for the complexities in a range of languages, see the contributions to Hirst & Di Cristo (1998), in particular the editors’ survey chapter. For instance, features such as pitch or nasality function subphonemically, phonemically, and morphologically, as well as in sentence domains, in various languages, so that a simple universally defined mapping does not work. Take just the “typical” intonation feature of pitch. This feature interprets categories in a number of domains:

- subphonemic: pitch perturbations caused by obstruent consonants, intrinsic pitch of vowels;
- phonemic pitch in South-East Asian and West African languages;
- inflexional pitch in West African languages (e.g. high tone for non-present tense);
- pitch patterns determined by word stress in Germanic languages, in derivation and compounding *EXport* vs. *exPORT*, *UMgehen* vs. *umGEHen*, *BLACKboard maker* vs. *black BOARDmaker*;
- utterance linking by (“final” , ‘sentence”) tone: *He’s /coming* vs. *He's \coming* (note -- not necessarily a simple question vs. statement opposition);
- focus by phrasal pitch accent: *The book with the TURQUOISE cover.* vs. *The BOOK with the turquoise cover*;
- operator scope by phrasal pitch contour extent: *He didn’t come because he was TIRED, ...* (i.e. he came) vs. *He didn’t COME because he was TIRED, ...* (i.e. he didn't come) – unless there is a higher level, more specific overriding context.

This multiple level structure lends itself easily to principled treatment in terms of defaults, which may be illustrated by using just one prosodic category, accent:

- Take the expression *ThirTEEN*. This corresponds to the “normal” , “default” word stress (in the usual sense of “default”), with *TEEN* realised by a pitch accent.
- This condition on accentuation can be “overridden” by a “stress clash” context: *THIRteen MEN*.
- This condition can be overridden, in turn, by a contrastive context: *I said “thirTEEN men” , and “thirTY women”* .
- This can in turn be overridden by a more extensive context: *I didn't say “THIRteen men and thirty women“, but “FOURteen men and thirty women“*.

The less specific contexts are overridden by more specific, more detailed contexts, corresponding to the general use of specificity as one of the usual conditions for deploying defaults to resolve conflicts. As usual in a closed world situation, the most general default case can be explicated monotonically as an abbreviation for the negative clauses in the following:

Accent is assigned to a syllable
IFF the syllable has lexical stress in the word AND
the word is the last lexical item in the utterance AND
the item is NOT ... AND
the item is NOT ... AND
the item is NOT ...

In the context of prosody it should be notex that there are meanings of “default” in the prosody literature which run counter to general usage in logic. If a debtor defaults on his payment, he does the opposite of what is expected, i.e. he does not pay. The term “default accent” was used by Ladd (1980) in this sense for the accent position in examples, in which *the book* is anaphorically “de-stressed”:

No, I haven't READ the book.

Conventional terminology would take the opposite view, and regard sentence final accent (“nuclear stress”) as the normal or default case, and anaphora as a special condition which overrides the normal case; this is discussed below. However, if defaults are not seen as all-or-none decisions, (which I will call the *mutatis mutandis*, or *other things being equal* notion of default), but as a hierarchy of less and less specific cases, Ladd’s terminology can be justified: accent falls on the final *available* lexical item. In these cases, the absolute rightmost item is anaphoric and thus not available. Consequently, accent is assigned to the rightmost available lexical item (which I shall call the *faute de mieux* notion of default, a special case of the former).

Prosodic inheritance and Integrative Lexicalism

But how do prosodic structures relate to a lexicon, in which entries may be more or less regular in terms of a hierarchy from the most general properties of lexical items to their most specific idiosyncrasies. In Figure 3, the overall structure of an integrated model for sign-based linguistic theory is outlined and motivated with respect to phonetic domains; see Gibbon (2000). Within this framework, studies have been made of a range of interesting morphological, phonological and prosodic systems, for languages as diverse as Arabic, English, German, Kikuyu (E. Africa), and Yacouba/Dan (West Africa), combining finite state modelling techniques for compositional aspects of the hierarchy with default inheritance networks for paradigmatic aspects; see Reinhard & al. (1991), Gibbon (1990), Gibbon (1991), Gibbon (1992a). The approach has been applied most extensively to the modelling of German inflectional morphology, including stress assignment, where a complete description (Bleiching & al. 1996) with a Prolog implementation has been developed, based on non-monotonic DATR prototype which was converted to a monotonic model by spelling out the defaults in the closed world of morphology, with a view to making it tractable for efficient reverse queries. For flexibility in querying, this version was in turn modelled in monotonically in cut-free core Prolog. Finally, the development turned full circle: a nonmonotonic version was implemented in Prolog for efficiency in paradigm generation for very large lexica (over 50000 forms). The implementations have been in use in the VERBMOBIL speech-to-speech translation project since 1994.

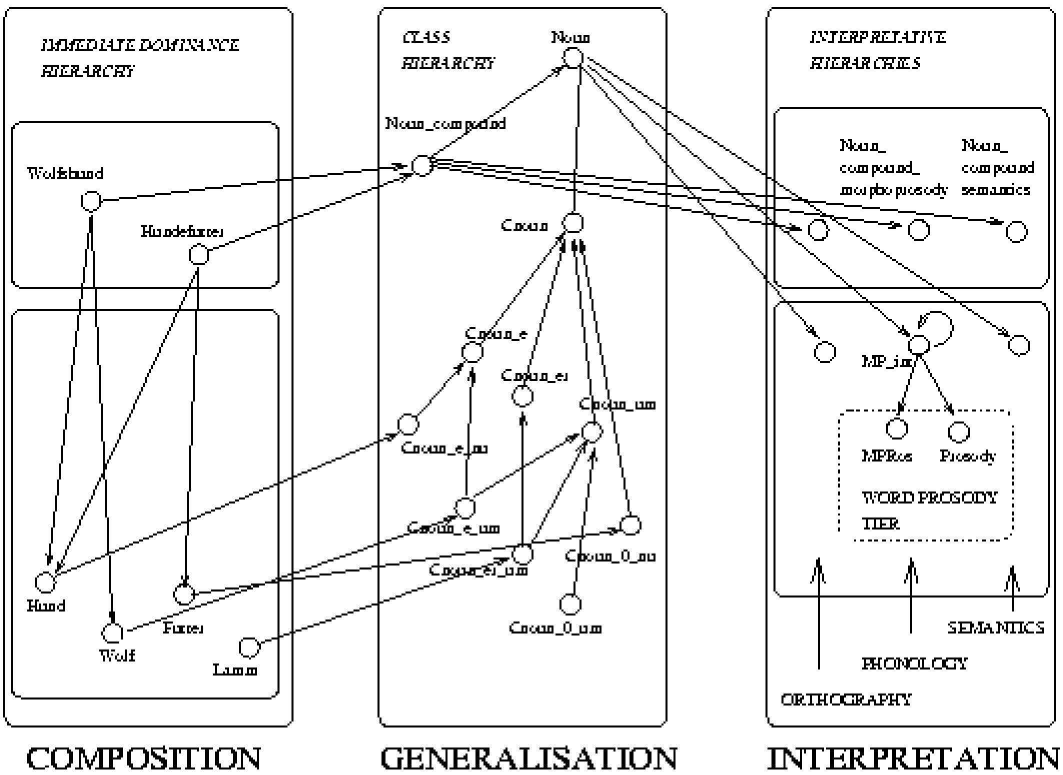


Figure 3: Prosodic Inheritance lexicon (for German nominal inflexion)

The basic format of Prosodic Inheritance Theory within the Integrated Lexicalist approach is to distinguish three components of sign structure:

1. COMPOSITION component: defines immediate dominance relations (downward arrows) in the lexicon and generically, over all linguistic levels from phonemes through to discourse idioms; roughly comparable with HPSG CATEGORY and DAUGHTER attributes.

2. GENERALISATION component: defines class hierarchies, "typically" as default hierarchies in order to be able to describe subregularities and idiosyncrasies (grammaticalisation and lexicalisation) compactly, but not necessarily; roughly comparable with HPSG type hierarchies.
3. INTERPRETATION component: defines mappings to semantics and phonetics (and visual modalities such as orthography, visible gesture); roughly comparable with HPSG PHON and SEM attributes.

These three components are orthogonal to the standard dimensions of phonology, morphology, syntax, etc., characterising architectures for language theories. The model is superimposed on the levels distinguished in Figure 3, and spells out the organisational aspects in more detail. of the model are shown in Figure 4. The levels of language organisation are not given in detail; the general outline is sufficient for present purposes.

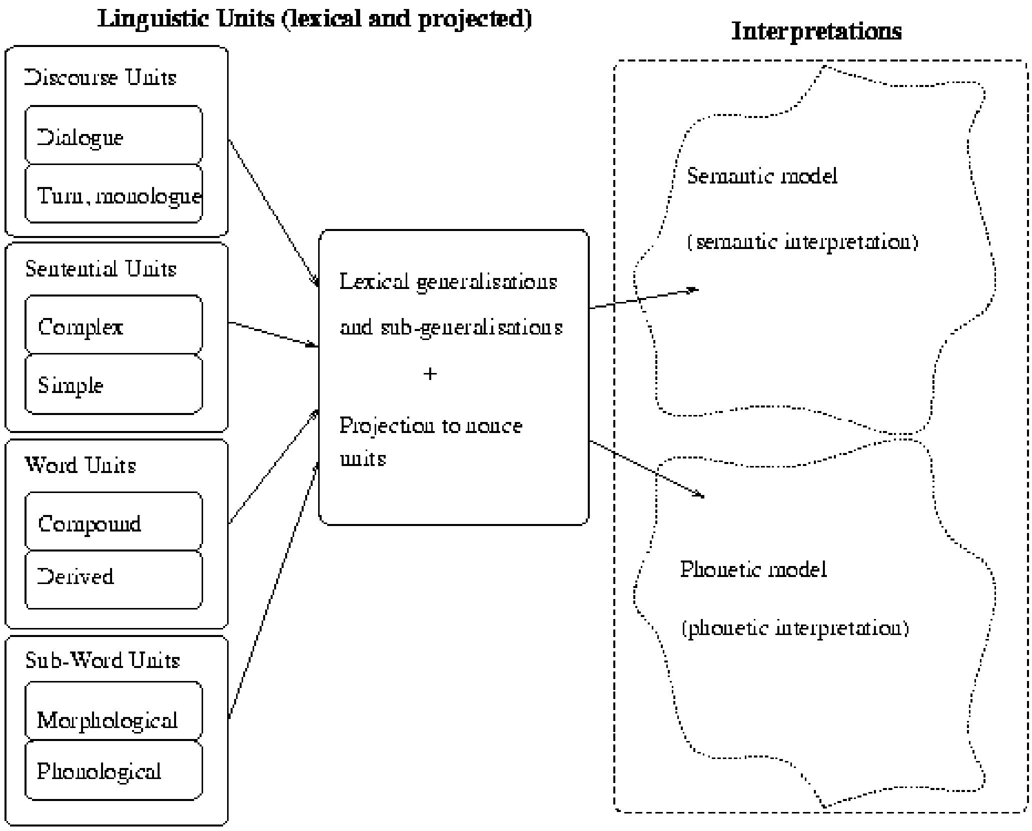


Figure 4: Integrated Lexicalist architecture.

The relevance of this model to discussion of the phonetic and phonological domain lies in the following points:

- Semantic and phonetic interpretation takes place at all levels of language sign structure; prosody (in the sense of stress, accent, intonation, etc.) is simply phonetic interpretation above the basic syllabic sub-word unit level.
- Both grammaticalisation and lexicalisation on the one hand, and productive creation on the other, take place at all levels.
- The joint phonetic-semantic domain introduced in previous discussion is simply the union of the phonetic and semantic domains in this model.
- The model represents "category time" in the sense of TMP (Time Map Phonology), and requires addition of the additional planes of "relative time" and "absolute time".
- In an operational (production and/or perception) version of the model, phonetic interpretaion must be mapped into the causal-temporal chain at all levels of language structure.
- In an interactive model, a structure of this kind is required for each partner in the interaction, with both

mapped into TMP structures and the causal-temporal chain.

The model is neutral with respect to the use of default reasoning; this depends on further theoretical assumptions. The relevance of the ranks of integrated lexicalist architecture is that apparent overrides of transparent compositional structures are easily explained in terms of grammaticalisation and lexicalisation relations, in which compositionality decreases over time; cf. the etymologies of English words such as *husband* or *woman*, where neither morphological, nor orthographic nor phonological levels are compositional any longer, with names like *Worcester*, in which the spelling is still partially compositional, but not the morphology or the phonology.

Syntagmatic defaults: default automata

Default constraints on compositionality, i.e. syntagmatic defaults, model structural preferences: preferential choices in non-deterministic situations (possibly probabilistically ordered) such as ambiguities, or in terms of complexity (e.g. hierarchies of preferred syllable structures such as CV, CVC, CCV CCVC; general structure-determining preferences such as sonority hierarchies). One way to model compositionality defaults is by means of default automata, which have become one of the most interesting aspects of computational phonology, combining work on defaults and preferences in phonological theory, and finite state techniques. In the following discussion a number of types of default automaton are discussed.

Fully specified full-string automata

In Figure 5 a transition network representation is given for a *fully specified full-string accepting automaton* for English syllable onsets. The transitions are labelled according to the SAMPA conventions (Wells 1989). A similar automaton on the same lines for German (based on an automaton due to Carson-Berndsen) is shown in Figure 6; phonotactic differences between the two languages are evident in the different topologies of the two transition networks.

The automata contain generalisations over similar transitions, expressed as sets, as well as overall distributional generalisations over sequences of lengths 2, 3 and 3, expressed as nondeterminism. The transitions can be spelled out and determinised, yielding a larger, but less perspicuous automaton.

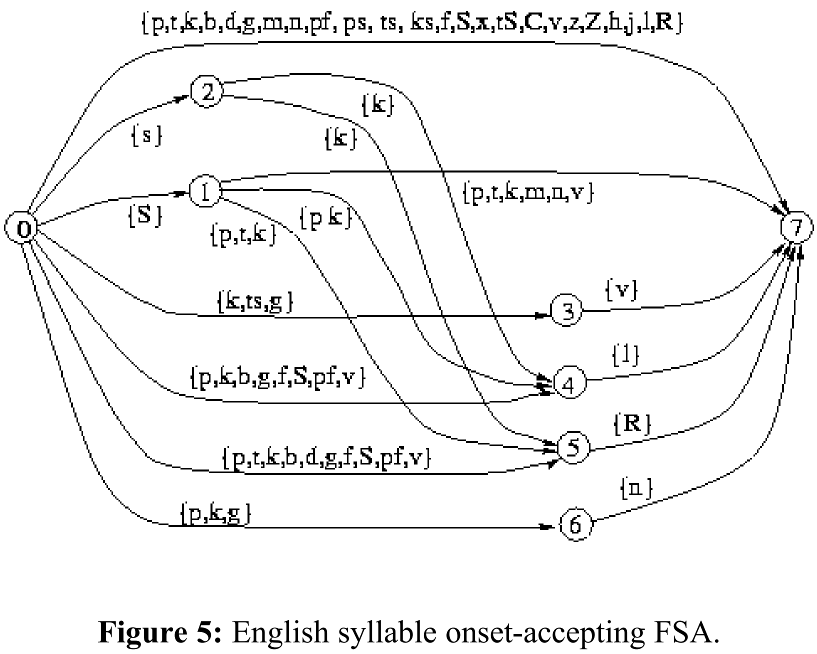


Figure 5: English syllable onset-accepting FSA.

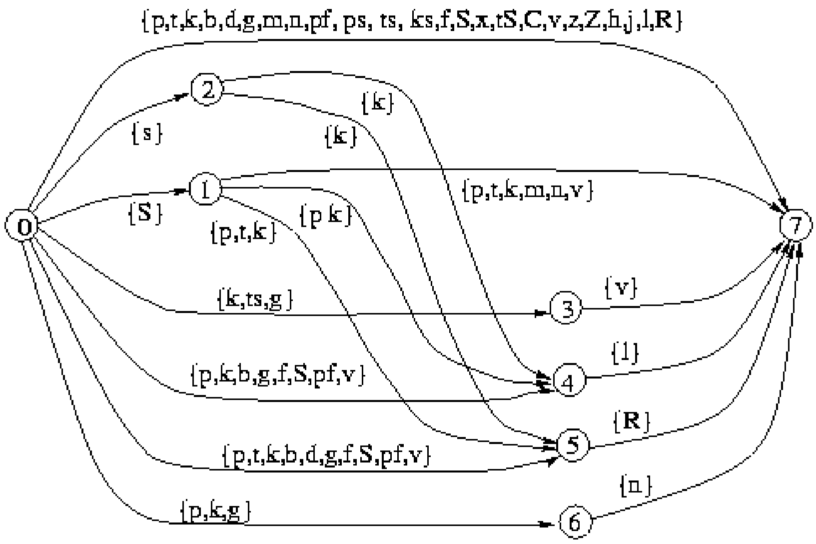


Figure 6: German syllable onset-accepting FSA.

Underspecified automata vs. underspecified models

Finite state automata with default properties have been used in a number of applications. The default properties stem from a variety of devices, ranging from the use of generalisations over the vocabulary, with variables (or underspecified feature structures) as transition labels, to the use of “elsewhere” (“ANY”, “OTHER”, etc.) variables on transitions as in the automata described in Martin Kay’s paper on metarules (1983), in Kimmo Koskenniemi’s two level rules (1983), in the default automata of Penn and Thomason (n.d., ca. 1994), and in the use of finite lookahead of varying degrees of specificity in prosodic inheritance FSAs (Reinhard & al. 1991) and Gibbon (1990).

Before starting out, we must distinguish between underspecified representations of automata, and automata used as underspecified linguistic representations, i.e. as underspecified models.

The sparse matrix representation of automata as a set of triples, i.e. as relation, is in general underspecified in the sense that no “sink state” or “fail state” is explicitly specified to which non-accepted vocabulary items lead, and that an unspecified transition is taken (by default!) as leading to a fail state. Empty cells in a full matrix representation, on the other hand, can be easily interpreted as specifying transitions to the “fail state”; in this sense the automaton has a fully specified representation (see Figure 7).

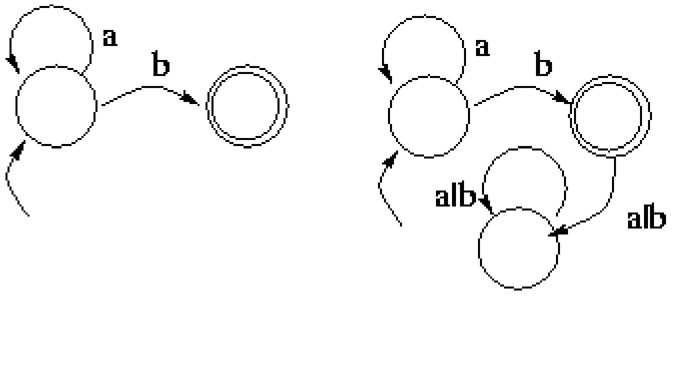


Figure 7: Graph representation of partially and fully specified automata.

Default automata as underspecified models

Automata as underspecified models, on the other hand, pick out certain properties of strings, e.g. classes of symbols, or substrings, and do not match the entire string. Effectively, they accept sets of strings based on specification of substrings, rather than single fully specified strings. Phonological rules are generally modelled by a specific kind of underspecified modelling automaton, which I will refer to as a stream processing or „freewheeling" automaton.

For a regular language *Lreg* (Type III in the Chomsky formal language hierarchy) there are accepting automata $A(Lreg) = <Q, q0\in Q, F\subseteq Q, \Sigma, \delta:Q\times\Sigma\rightarrow Q>$, where Q is a finite set of states, $q0$ is the initial (start) state, F is the set of final states, Σ is a finite vocabulary, ϑ is the transition function from state to state, and $Lreg\subseteq\Sigma^*$; Σ^* (the „Kleene closure" of Σ) is the infinite set of all strings, including the empty string, formed by concatenating the elements of Σ. In most applications, automata are conventionally designed to process complete strings $w\in L$. In many applications in computational linguistics, for example two-level morphology (Koskenniemi 1983), as well as in UNIX text stream editing tools, tokeniser (lexical analysis) components of compilers, on the other hand, automata are designed to spot and process regular substrings rather than complete strings, i.e. $\{v \mid v, xvy \in Lreg\}$.

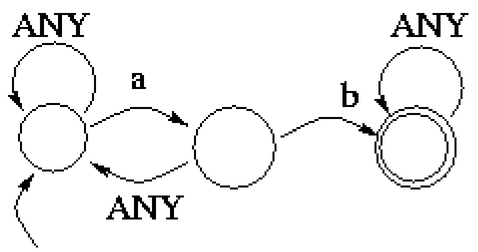


Figure 8: Stream processing „freewheeling" default automata.

These 'stream processors", which process arbitrary substring occurrences in strings of arbitrary length, can be represented as cyclic automata with „default" transitions, i.e. transitions labelled with state-specific variables which range over all elements of the vocabulary which do *not* explicitly label transitions leaving a given state. These automata freewheel" through the string, cycling on the start node, until an explicitly specified element of the vocabulary is accepted. If at any point an input symbol is not accepted, a default transition returns to the start node, at which point the freewheeling starts again. The automaton in Figure 8 shows an automaton which accepts string containing a substring ab. There are other varieties of automaton involving obligatoriness and optionality, and in particular involving finite state transducers, but the principle remains. Labels such as =„, ANY" (or the point in UNIX tools, e.g. /.*ab.*/ for the automaton in Figure 8) are often used to represent such variables. Incidentally: this is one reason why UNIX tools, including scripting languages such as Perl, are extremely useful prototyping tools in computational phonolgy. An extremely detailed account of finite automata and their uses in computational phonology, natural language processing and speech technology is given by Jurafsky & Martin (2000).

The freewheeling automata are also underspecified, but evidently it is no problem to map them to fully specified automata, given the vocabulary specification for the automaton, exhaustively spelling out the default labels: we have a finite set, a closed world, and therefore the default device is simply an abbreviation convention. Automata (more particularly: transducers) of this kind are easily adaptable to modelling phonological rules.

For instance, an automaton for assimilating any /m/ to the place of articulation of a following /b/ (i.e. to /n/), assuming at least one symbol intervening before the next occurrence of /mb/, is shown in Figure 9; generalisation to a feature-based automaton or to an automaton with autosegmental transition conditions is straightforward. The following conventions are used: ANY transduces any default symbol to itself; any symbol standing along transduces to itself; for any symbol pair separated by a colon, the left hand symbol transduces to the right hand symbol. It is in this sense that defaults are used in order to define underspecified models for phonological rules; the automata themselves are fully specified.

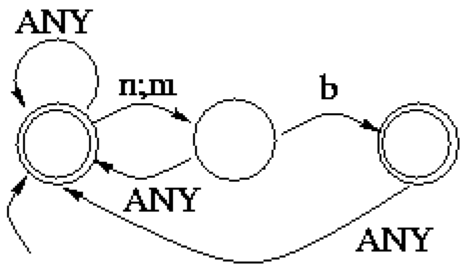


Figure 9: Stream processing transducer for place assimilation.

Penn-Thomason default finite state machines

Penn and Thomason (n.d., ca. 1994) propose a type of default automaton, which they term DFSM and which I will call PTM (Penn-Thomason Machine) in order to avoid confusion with the use of „D" in this context to mean deterministic.

The PTM is introduced as a formalisation of two-level morphology rules which explicitly incorporate specificity ordering (corresponding to the „elsewhere condition") over contexts:

> DFSM's extend FSM's (specifically, finite-state transducers) so that transitions can be context-sensitive, and enforce a preference for the maximally specific transitions. The first change allows phonological rules to appear as labels of transition arcs in transducers; the second change incorporates the elsewhere condition into the computational model.

In a footnote it is pointed out that

> The elsewhere condition is bulit into an implementation of the TWOL rule compiler; ... But on this approach, default reasoning and the elsewhere condition are not employed at a level of computation that is theoretically modelled; this reasoning is simply a convenient feature of the code that translates rules into finite state automata.

The PTM is like the type of default automaton already discussed to the extent that, like them, it applies to substrings rather than complete strings; more precisely, it applies to sets of strings sharing a substring, and in order to do this, transitions are labelled with encodings of phonological rules. Additionally the PTM defines specificity orderings over sets of contexts for phonological rules, a feature which enables it to model the *elsewhere condition* which is a prominent features of rules in classical Chomsky-Halle style generative phonology. The basic idea is this:

1. A *replacement* is a pairs of symbols over a vocabulary (including the null symbol); the leftmost element of a pair is *underlying* (U), the rightmost element is *surface* (S).
2. The basic kinds of object are letters, replacements, strings of replacements, and variables over these.
3. A *US-string* is a string over the set of replacements consists of a U-string and an S-string.
4. A *rule* is a pair of a set of pairs of US-strings (note: a set), and a replacement type.
5. A pair of US-strings, e.g. <**x**,**x**> for **x**=<x,x'> and **y**=<y,y'> together with a replacement **l**, **l**=<l,l'>, constitutes a *contextualised replacement* <**x**,**l**,**y**>.

6. A contextualised replacement satisfies a rule if the left and right contexts are in the US-string set of the rule, and the replacement is in the replacement set of the rule.
7. The claim is made that pairs of regular sets (as in 2L morphology) are not required for encoding contexts, but that a finite set of pairs of strings is sufficient.
8. A string *satisfies* the left (right) half of a context if its right (left) matches one of the strings encoding that half.
9. Notation: **U**___**V**, with $\mathbf{U},\mathbf{V} \subseteq$ US-strings refers to the context encoded by $\{\langle \mathbf{u},\mathbf{v} \rangle \mid \mathbf{u} \in \mathbf{U}, \mathbf{v} \in \mathbf{V}\}$.

Example:

Rule encoding: $\langle \{\langle \mathbf{c},+ : 0 \rangle \mid \mathbf{c} \in \mathbf{C}\}, \{y : i\} \rangle$
Rule notation: $y \rightarrow i$ / **C**+:0
Rule description: Replace *y* by *i* before a morpheme boundary and after a constant US-consonant, i.e. after *l* : *l*, where $l \in \mathbf{C}$

The rule is actually a spelling rule rather than a phonological rule. The set of underlying-surface consonant pairs is **C**; the US-string +:0 means that the underlying morpheme boundary has no correlate in the surface string. The rule describes cases such as *lady-ladies*, *lazy-laziness*.

The notion of default is introduced using the relation of *context specificity*:

> A context C is more specific than a context C' iff C is a subset of C'.

A context is a pair of sets of US-strings, and a *context type* is a set of such pairs. An indexed US-string is what was earlier called a contextualised replacement.

Finally, DFSMs are described as follows:

> A DFSM's transitions are labelled with finitely encodable rules rather than with pairs of symbols. Moreover, nondeterminism is restricted so that in case of conflicting transitions, a maximally specific transition must be selected. The critical definition is that of *minimal satisfaction of an arc by an indexed path, where an indexed path represents a DFSM derivation*, by recording the state transitions and replacements that are traversed in processing a US-string.

DATR automata with parametrisable context length

Default automata which have the features of Penn-Thomason-Machines, PTMs, and, like them, are also directly encoded as automata rather than via other notations, have been developed using the DATR formalism (Gibbon 1990, 1991, 1992a). The default mechanism of DATR is based on the specificity of string (path) prefixes: given two paths which match prefixes of a given sequence, the longer path overrides the shorter one. The shortest possible („match all") path is the empty path. Consequently, when DATR is used to implement automata paths of different lengths can be used to model lookahead. The /mb/ assimilation rule would look like this, using iteration with a DATR variable `$any` as a freewheeling variable:

```
Assimilation:
<m b>      == n <b>
<$any>     == $any <>
<>         == .
```

In DATR procedural semantics, if a left-hand side path matches the prefix of a sequence, then any right-hand side paths are concatenated with the remaining non-matching suffix and then evaluated. The meaning of RHS `$any <>` is „output the value of the variable `$any` and concatenate this with the result of evaluating a query at the same node, but with an empty path concatenated with the remaining suffix". Similarly for the RHS `n <b>`, except that for the new query at the same node, the path `<b>` is concatenated with the remaining suffix. In this specific case, this RHS could also have been formulated `n b <>`; in the general case, however, this would preclude matching `<b>` again in a subsequent context, if required.

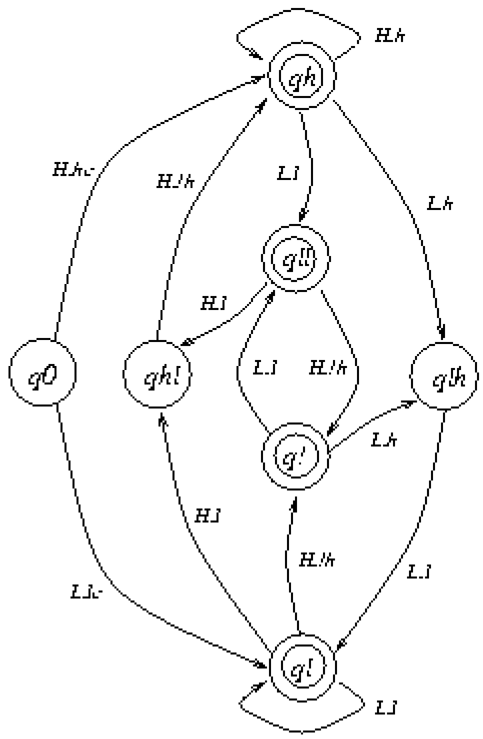


Figure 10: Baule tone sandhi automaton.

A fully specified automaton model for a „real-life” situation is a model I developed for tone alternations in the West African (Kwa) language Baule (Gibbon 1987), cf. Figure 10. The cycles represent different metrical (rhythmic, iterative) properties of the tone system: the same-state cycles (epicycles) represent sequences of high or low tones, the large outer cycle represents a „terrace” consisting of a high sequence demi-terrace and a low sequence demi-terrace (or vice versa), and the inner cycle represents a special case of lookahead.

One possible model in DATR, using ‘shortest path defaults’ and ‘longest path overrides’ to express specificity of contexts, formalising „lookahead” as path overriding, is as follows:

```
% Start:
Baule_q0:
  <h>   == hc Baule_q1:<>
  <l>   == lc Baule_q2:<>
  <>    == .

% After a high tone:
Baule_q1:
  <h>   == h <>
  <l l> == h l Baule_q2:<>
  <>    == Baule_q2 .

% After a low tone:
Baule_q2:
  <l>   == l <>
  <h h> == l !h Baule_q1:<>
  <h>   == !h Baule_q1:<>
  <>    == Baule_q0 .
```

The lookahead function is modelled by means of the path override semantics: the more specific <h h>, representing a sequence of two high tones, is preferred to <h>. This use of defaults has two conventional properties, first the use of a specificity relation, here DATR path override; second, the use of defaults as a compaction operation, reducing a finite set of states and transitions to a smaller set.

Default automata with constraints over natural classes

In Gibbon (1985), partially reproduced in Carson-Berndsen (1998), I presented a default finite state automaton as an underspecified model for English syllable phonotactics (Figure 11).

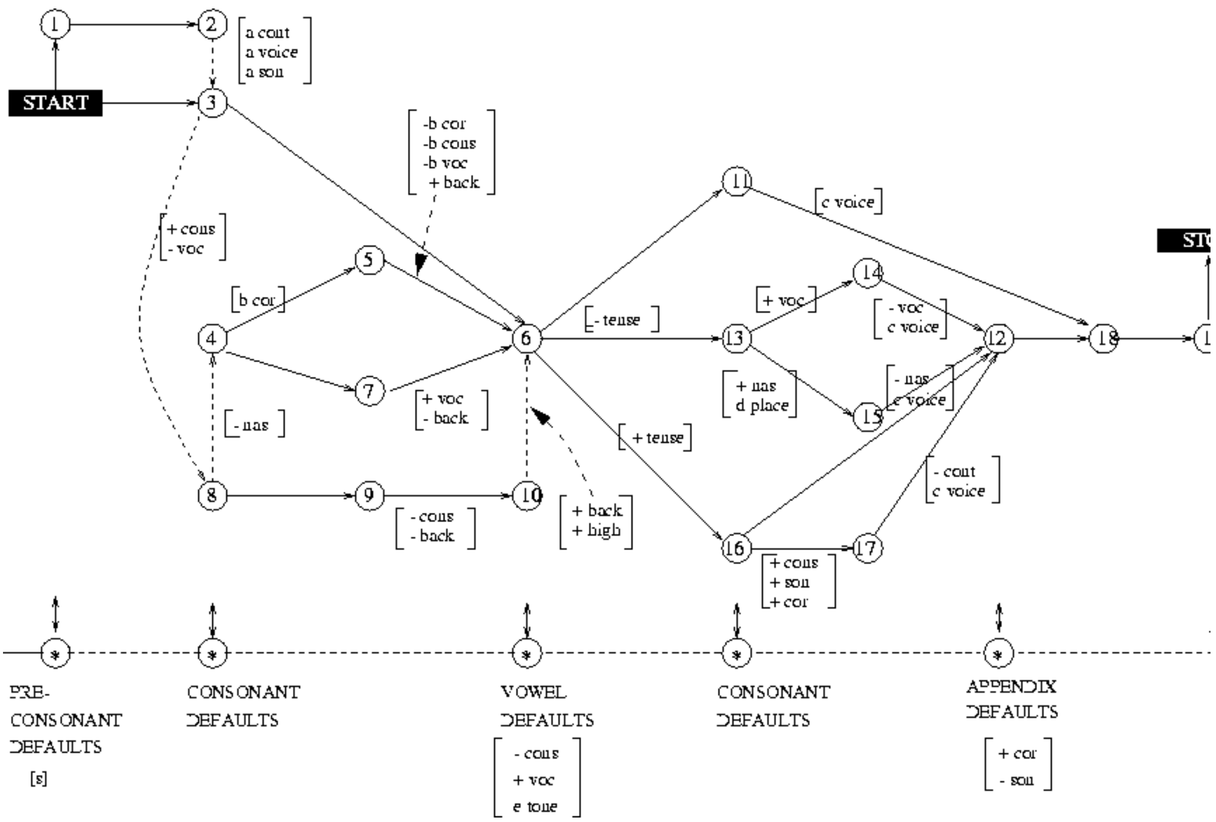


Figure 11: „Online default automaton" for English syllables (Gibbon 1985).

This model needs cleaning up, formally, and cannot be described in detail here; it provides an early illustration of compositionality default generalisations. It differs from other currently known types of default automaton in the following ways:

- The transitions specify sets of symbols in terms of feature structures.
- Assimilation and dissimilation constraints between neighbouring transitions are modelled by variables over feature values (the „Greek variables" of generative phonology).
- A precedence ordering („left-right", „before-after") determines default specifications for feature values: defaults are defined for the onset, and carried forward *unless explicitly specified otherwise*, then for the nucleus, etc.
- The override relation thus mirrors precedence in two ways: first, within a default domain (e.g. onset), explicit specifications override the defaults; second, at the beginning of a new domain (e.g. nucleus) a new set of defaults overrides the old set.

This automaton combines several features of the previously discussed types:

1. It combines transition underspecification with lookahead.
2. It is incremental, with initial default conditions overridden by later conditions.
3. The two forms of override model both intra-domain and inter-domain overrides.

The inter-domain overrides can be re-interpreted monotonically if the five override points are simply regarded as the beginnings of separate concatenated automata. This does not apply to the intra-domain overrides.

In comparison with the straightforward syllable onset automata already outlined, the greater degree of generalisation offered by left-right default specifications can clearly be seen in the reduced set of nodes and transitions.

Tentative classification of default automata

From the preceding discussion, the following classification of default automata arises:

1. Default (sparse matrix, relational) representations of automata.
2. Stream processing („freewheeling", „filter") automata as default models:
   a.ANY-loop automata: automata with default („ANY") labels for stream traversal loops (cf. two-level morphology);
   b. Penn-Thomason Machines (Default Finite State Machines): automata with string set labels ordered by a specificity condition (formalisation of ANY-loop automata).
3. Full string accepting default automata:
   a. DATR default prefix automata: with lookahead scope modelled by path prefix length with longer matching prefixes counting as more specific than, and overriding, shorter prefixes (DATR);
   b. Online feature override automata: with default class (feature bundle) labels and „online" precedence overriding.

The Penn-Thomason approach to formalising automata appears very promising; if it could be generalised to cover the other kinds of automaton described here then one might hope for a well-defined parametrisation which can be viewed as an „automaton space", with different automaton types suited to different purposes, but reducible to a common core formalisation.

Perhaps the most interesting question is that of *phonological learning*. In terms of the present survey, this would amount to the following tasks for formal induction or machine learning:

1. Induction of a context-determined segment classification.
2. Identification of characteristic phonetic properties of these classes.
3. Induction of generalised automata on the basis of the classification.
4. Induction of generalised automata on the basis of assumptions about feature specification defaults with plausible procedural semantics (e.g. linear precedence determined).

Almost an afterthought: Bayesian ranking and Hidden Markov Models

The standard computing devices used in speech recognition are Hidden Markov Models, which can be described as finite state automata with probabilities attached both to transitions and elements of the output vocabulary. A discussion of statistical methods, and of Hidden Markov Models (HMMs) n particular, cannot be more than an aside in the present context. I would just like to note the following:

- It is hardly surprising to a computational phonologist that HMMs are so successful in speech recognition in view of the suitability of finite state transducers for modelling and implemention in phonology.
- As finite state transducers HMMs fit well into the current landscape of finite state linguistic technologies in general.
- HMMs are transducers, and are thus subject to the usual restrictions on operations over transducers.

- In particular, HMMs are non-deterministic, in that for a given input they produce more than one output.
- Transitions and outputs are both statistically weighted, potentially leading to multiple paths for a given output.
- Statistical weights do not only determine transition probabilities, i.e. cooccurrences, but also correspond to one kind of criterion for linguistic markedness.
- HMMs are conventionally used as *acoustic models*, i.e. for inputs derived by transformation from speech signals, and as such represent the phonetics-phonology relation.
- The outputs („hypotheses“) of the acoustic models are weighted linguistic units (e.g. phonemes, words), i.e. sets of linguistic units with an ordering defined over them on the basis of Bayesian quantitative evidence.
- A given prioritising ordering may be overridden by evidence from other components, e.g. *language models* which assess the probability of the occurrence of given outputs in the given linguistic context.
- The prioritising property of HMMs (and other search components in automatic speech recognition systems) lends them non-monotonic properties.

These features of HMMs (which may be generalised to other devices) clearly point to an affinity with many areas discussed above, in particular non-monotonic reasoning and default finite state automata. See Jurafsky & Martin (2000) for further discussion.

There is a further procedural (in the present context perhaps better: operational) aspect of the deployment of HMMs which needs to be further considered, namely that they are typically used in speech *recognition*, textual *analysis*, and map surface representations to sets of hypotheses about underlying representations. In contrast, linguistic rules are typically formulated as *generative functions* in a sense not originally intended by the inventors of generative grammar: they map underlying to surface reprsentations. It seems to me that part of the conceptual problem underlying the use of defaults is exactly this dilemma: linguists want nice, clean *functions*, but there are so many factors involved that outputs are not simple functions of particular autonomous components or principles, and involve complex interactions between many systems.

A challenge for non-monotonic reasoning as applied to phonology is therefore a general epistemological one: Can linguistics be given intellectual tools to enable linguists to cope with local non-functionality which lands them with local optima with untidy edges, and lead them further in the hope that somewhere in the distance a global function with a global optimum really does exist?

Paradigmatic defaults: phonological feature theory

Paradigmatic defaults are concerned with similarities among linguistic units with respect to some „normal” representative of a class; they typically affect markedness relations in oppositions and between values of features in phonology.

Features and value-changing rules

The history of phonology and linguistic phonetics has been characterised by one formal device in particular: features. Features are Janus-faced:

- on the one hand, they are ascribed phonetic properties (often articulatory, sometimes acoustic);
- on the other, the property of being *distinctive*, i.e. of distinguishing one phoneme (ultimately, one word) from another.

Although the basic intuition that features are associated with phonetic properties, a 50 year hunt for articulatory, acoustic and auditory „phonetic correlates” has produced many generalisable empirical results, but results which are not fine-grained enough to be routinely operationalised, for instance, in high quality automatic speech synthesis or in accurate automatic speech recognition. The most natural synthesisers today are still concatenative and derived from real corpora (diphone synthesisers and related devices), though there are signs that parametric synthesizers may be catching up. Likewise, with automatic speech recognition, the best recognisers are also still concatenative, based on phonemes, phoneme sequences, syllable constituents, etc., but there are signs that „autosegmental” parallel channel recognisers are emerging.

One standard notation for features is [- continuant, - nasal, + voice]. This defines the set of voiced stops (plosives), i.e. /b, d, g/ in English. Note that the features are theory-specific, although many attempts to formulate a consensus on feature universals have been made. The feature specifications (values) are Boolean, that is the values are not independent empirical properties, as in many linguistic applications of attributes, where attributes simply provide an ordering context for mutually exclusive empirical properties. Features and feature specifications are only interpretable as pairs, not as atoms: <-, continuant>, <-, nasal> and <+, voice>.
Phonological rules can be regarded as constraints over feature specifications, formerly expressed by means of the „Greek variable convention“, replaced over two decades ago by autosegmental lattices. However, the older conventions provide convenient illustrations of phonological rules as default notations. For example, voicing assimilation of obstruents (stops and fricatives) between vowels (or other voiced items) may be expressed as follows:

$$[+\ \text{cons}] \rightarrow [\alpha\ \text{voice}]\ /\ [\alpha\ \text{voice}\ ]\ \_\_\ [\alpha\ \text{voice}]$$

This is apparently an abbreviation for two rules, as the variable α ranges over two values, {+, -}:

[+ cons]	$\rightarrow$	[+ voice]	/	[+ voice]	____	[+ voice]
[+ cons]	$\rightarrow$	[- voice]	/	[- voice]	____	[- voice]

but actually an abbreviation for 4 rules, because of the underspecification of the consonantal segment with respect to voicing:

[+ cons, + voice]	$\rightarrow$	[+ voice]	/	[+ voice]	____	[+ voice]
[+ cons, - voice]	$\rightarrow$	[- voice]	/	[- voice]	____	[- voice]
[+ cons, + voice]	$\rightarrow$	[+ voice]	/	[+ voice]	____	[+ voice]
[+ cons, - voice]	$\rightarrow$	[- voice]	/	[- voice]	____	[- voice]

corresponding to the following mapping formulated directly between strings of feature bundles:

[+ voice]	[+ cons, + voice]	[+ voice]	$\rightarrow$	[+ voice]	[+ cons, + voice]	[+ voice]
[+ voice]	[+ cons, - voice]	[+ voice]	$\rightarrow$	[+ voice]	[+ cons, + voice]	[+ voice]
[- voice]	[+ cons, + voice]	[- voice]	$\rightarrow$	[- voice]	[+ cons, - voice]	[- voice]
[- voice]	[+ cons, - voice]	[- voice]	$\rightarrow$	[- voice]	[+ cons, - voice]	[- voice]

The first and the fourth mappings are vacuous, as nothing changes, a fact which is not evident from the original formulation of the phonological rule, but which corresponds to the default case of voicing „harmony” in adjacent segments; values are only overridden in the case of dissimilar values. The notation therefore clearly involved unspoken assumptions about defaults; it is easily spelled out as a monotonic relation as shown, corresponding to two level transduction rules in computational phonology.

A further important non-monotonic element remains: the *feature changing* property of this kind of phonological rule. In the non-vacuous case, a feature is changed; this is non-monotonic, because a representation which was licensed as well-formed is „negated“: an opposite specification is licensed. This kind of rule illustrates why, in terms of the domain discussion, „phonology is different“. But can this non-monotonic property be factored out, or is it a necessary feature? The following points are central to this discussion:

- As pointed out in the discussion of phonetic domains, the feature of non-monotonicity only rears its head if the features on the LHS and the RHS of the rule are regarded as being *in the same domain*. If this assumption is made, then it looks as though lexical axioms of the language (i.e. givens about the phonological structure of lexical items) are being replaced, introducing non-monotonicity in the strict sense.
- However, if the rule is regarded as a transducer which actually does relate structures in different domains, either abstractly or within the causal-temporal phonetic chain, then this problem disappears, and we simply have a mapping from one subdomain to another.
- The mapping between phonological and phonetic subdomains can be modelled, as is well known, by finite state transducers.
- There still remains the problem of epistemologically or causally and temporally relating the stages in the transducer cascade, the „How abstract is phonology?" question raised by Paul Kiparsky 30 years ago.
- In a different, formal context, a related question has been raised in computational phonology: if finite state transducer cascades are allowed to apply to their own output domains, the result is Turing equivalent (Johnson 1972), Kaplan & Kay (1994).

I take these aspects of phonological representation, apparently only loosely related, to be pointers to aspects of the human language faculty which are still in need of intensive research, in linguistics, psycholinguistics and in logic. As a first step, default unification (cf. Bouma 1992) and subsequent work by Copestake, Carpenter, Lascarides and others) comes to mind as a technique for modelling the kind of relation expressed by the feature-changing rule; the rule format and operation is essentially identical to that of lexical rules in mainstream HPSG, and just as controversial.

Privative oppositions, markedness, weighted attribute values

The feature specification model was modified by Chomsky and Halle (1968) and related more closely to empirical observations in their *theory of markedness*. Markedness is a prioritising of binary feature values, often sensitive to neighbouring context. For example:

[u voice] → [- voice] / [+ consonantal]
[+ voice]

This is a default rule - with one elsewhere condition - as with other rules discussed so far. The empirical question is, of course, how to determine which of the contexts is the more specific. The answer given for the introductory example of English plurals, i.e. applicability to more segment types, can apply here, too. But in fact, several kinds of argument are used:

1. Frequency of occurrence in a corpus (token frequency).
2. Frequency of occurrence in an inventory (type frequency; see above).
3. Frequency of occurrence in the languages of the world.
4. Frequency of occurrence in structural contexts (neutralisation).

The last of these is the most interesting, and actually the original motivation given by Trubetzkoy for the subclass of *privative oppositions*, as opposed to *equipollent oppositions* (1939). The standard example of neutralisation is final devoicing in languages German, Dutch, Russian, and other languages, a rule which does not hold in English. Final devoicing means that a voiced obstruent is voiceless in final position:

[+ voice] → [- voice] / [+ consonantal] #

This means that the specification of the voice feature is "changed" from + to - in a consonant before a boundary (whether syllable or word boundary is not a concern here). There are complications, for instance entire consonant clusters may be affected, but this is not at issue here.

There is an interesting consequence of final devoicing for morphological paradigms in which the stem is sometimes final and sometimes not, by virtue of having a suffix. So, for example, in the word *meiden*, with the preterite third person singular and plural forms in German *er mied - sie mieden*, /eɐ mit/ - /zi midən/. The stem-final alveolar plosive is voiceless in the singular, because the final devoicing rules applies, and voiced in the plural, becase the final devoicing rule does not apply.

Contrast this with the schwa-less present singular imperative and present plural forms of the word *mieten*: *miet - sie mieten*, /mit/ - /zi mitən/. The singular forms corresponding to the orthographies *mied* and *miet* are identical: /mit/ In cases like this, *the opposition is neutralised*, and the most frequent form in the paradigm (also the simplest to specify), in this case the voiceless form, is said to be the unmarked form. Underlying this notion of markedness is a phonetic assumption: with unmarked, neutralised forms, a phonetic property is genuinely *missing*; the opposition is otherwise lexically *marked* by a contrast between the presence and the absence of this feature.

The default reasoning aspect of the markedness relation is that in lexical representations, all values are unmarked UNLESS otherwise specified. So [+ consonantal] would always be [u voice, + consonantal] unless explicitly marked in the lexicon as [m voice]:

$$[\text{m voice}] \rightarrow [\text{- voice}] \;/\; [\text{+ } \overline{\text{consonantal}} \quad]\ \#$$

Rule ordering

The phonological rules discussed so far have been ordered on the grounds of specificity. However, there are other kinds of phonological rule, and these are ordered on different ground.

One important ordering principle is the *phonological cycle*, which determines word-stress and the vowel modifications which depend on stress. The phonological cycle is a compositional interpretation principle; the formulations are invariably procedural. The Nuclear Stress Rule is perhaps the most startling in its elegance. It is a function which maps strings of phrasal constituents into strings of numbers representing relative stress. This rule is fun: there are numerous algorithms which calculate it, and I have invented a new one for the occasion. So here is a novel algorithm to calculate the NSR function (for an assignment, see Figure 12).

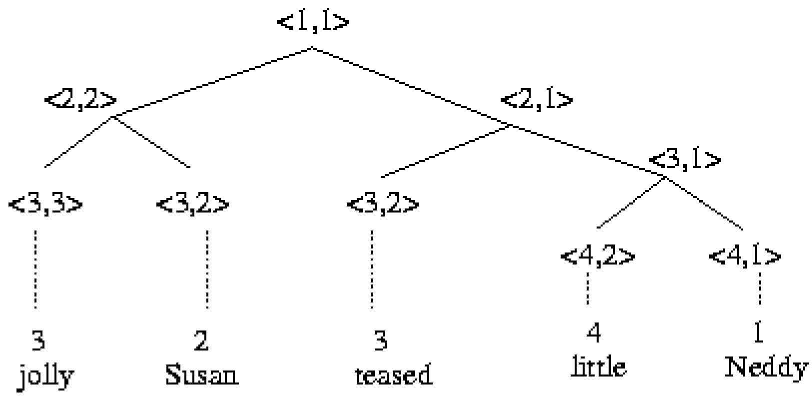


Figure 12: Nuclear stress assignment.

Nuclear Stress Tree Encoding:

1. Assign a value pair <1,1> to the root node.
2. Recursively

3. assign a right daughter node the values <*n*+1,*m*>, for a mother node valued <*n*,*m*>; if a leaf node, output *m*.
4. assign a left daughter node the values <*n*+1,*m*+1>, for a mother valued <*n*,*m*>; if a leaf node, output *n*+1.

There are many other formulations of this function and, as in other cases, there are complications, but these are not at issue here. Interestingly, the is a corresponding rule for word-internal stress assignment, the Compound Stress Rule, which (at the present level of granularity) is the mirror image of the Nuclear Stress Rule. The relevance of NSR for default reasoning in phonology has already been informally introduced in the discussion of the prosodic examples such as *No I haven't READ the book* and *THIRteen MEN*.

There are many other kinds of rule ordering, and rule application principles for repeated applications of the same rule (for instance, left-right, right-left, domain specific, across the board) with functional inter-rule relations of feeding and bleeding, counter-feeding and counter-bleeding.

In a procedurally rather messy situation of this kind, declarative approaches such as Edinburgh Declarative Phonology came as a refreshing change in the late eighties and early nineties: in a purely monotonic declarative phonology, ideally there are only representations and generalisations over representations (i.e. constraints), and the only rule in phonology would be *modus ponens*, and everything else would fall out automatically by *modus ponens* from axioms about phonological relations. Further generalisations using default preferences would require an additional principle of ordering generalisations, for instance by specificity, observed frequency (or probability) etc.

Vennemann's Preference Laws

Very closely related uses of monontonicity and default notions in general are found in *preference theories*. Preference theories are not concerned with the issue of simplifying descriptions to capture more and more empirical generalisations, but with the issue of substantive linguistic universals, involving dimensions of linguistic explanation relating to

- explaining similarities and differences between languages in general;
- capturing the relative complexity of structure in different languages;
- describing possible relations of realisation between lexically generalised representations and observable phonetic representations;
- explaining learnability.

In a number of noteworthy publications Vennemann discusses a number of “laws” for syllable structure (1988 p.1) which combine syntagmatic with paradigmatic preferences:

> These laws specify the preferred syllabic patterns of natural languages as well as determine the direction of syllable structure change.

The notion of preference is put into the following context (p.1):

> My conception of preference laws differs from most approaches to linguistic naturalness by characterizing linguistic structure not as good or bad (natural or unnatural, unmarked or marked), but as better or worse.
>
> The basic schema for preference laws is (p.1):
>
> „X is the more preferred in terms of (a given parameter of) syllable structure, the more Y“, where X is a phonological pattern and Y a gradable property of X

Preference Law Theory anticipates the foundations of *Optimality Theory* (see below), which, however, does not acknowledge Vennemann’s pioneering work. Vennemann’s approach to optimality (“improvement”) is diachronic:

For instance, every syllable structure change is an improvement of syllable structure as defined by some preference law for syllable structure. If a change worsens syllable structure, it is not a syllable structure change, by which I mean a change motivated by syllable structure, but a change on some other parameter which merely happens also to affect syllable structure.

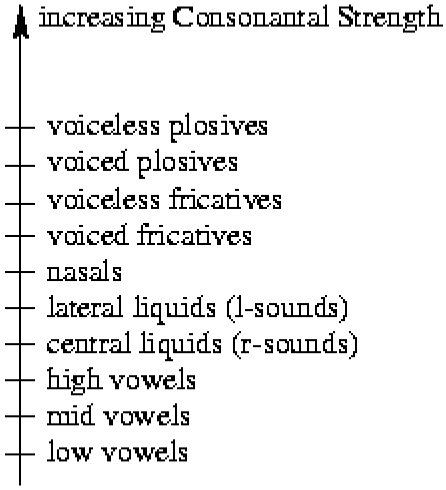


Figure 13: Vennemann's scale of Consonantal Strength.

In Venneman's Preference Theory, non-monotonicity plays a dual role:

1. The preference orderings defined in the individual laws.
2. An ordering related to *degree of unimpeded (voiced) air flow*, „Universal Consonantal Strength", shown in Figure 13). There have been many proposals for scales such as this (the inverse being 'sonority"), but the point here is not whether Vennemann's ordering is superior or inferior to those suggested more recently. The point here is not substance but the form of the model, and Vennemann's approach is empirically and formally by far the most carefully thought out in terms of the consequences of this kind of ordering for an explanatory phonological theory in general (not only for language change in particular), rather than just as a descriptive device for specific problems.

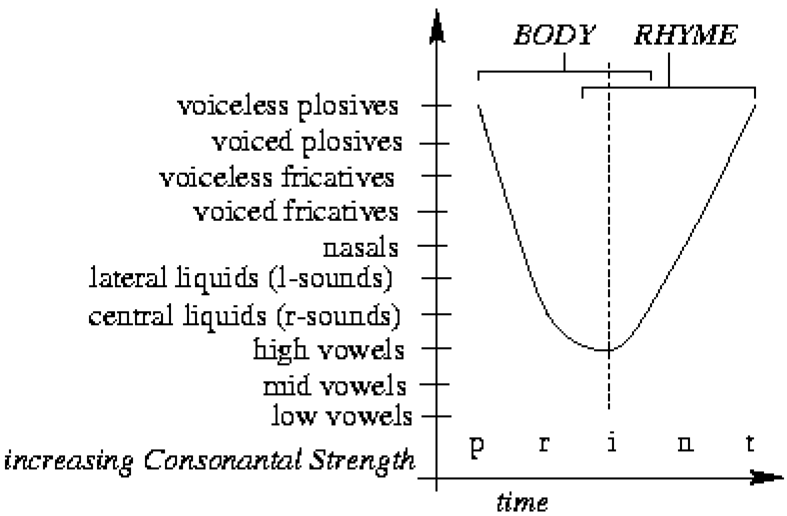


Figure 14: A strongly monotonic syllable (with strongly monotonic BODY and RHYME)..

Vennemann defines *strong monotonicity* and *weak monotonicity* (in the basic sense of the terms, not in the sense of non-monotonic reasoning) as a property of syllable parts with reference to this ordering (p. 9):

> The concept of Consonantal Strength allows us to define the concept of **monotonicity** for heads, bodies, codas, rhymes, and syllables. Heads and bodies are called weakly monotonic if no rise, and codas and rhymes, if no fall of Consonantal Strength occurs in them. Heads, bodies, codas, and rhymes are called **strongly monotonic** if hey are weakly monotonic and do not contain sound occurrences of equal Strength. **A syllable is called weakly (strongly) monotonic** if both its body and its rhyme are weakly (strongly) monotonic. A strongly monotonic syllable is called a **core syllable** is called a *core syllable*.

Examples of core syllables are *two*, *tip*, *trick*, *trunk*. This notion of monotonicity may be interpreted as the markedness kind of default, but it provides the main reference scale for non-monotonicity in the preference laws.

Buthat are the syllable preference laws? Vennemann discusses the following, pointing out that there must be many more laws (he mentions the Shell Law, the Body Law and the Rhyme Law):

1. Preference laws for individual syllables
    1. **Head Law** A syllable head is the more preferred: (a) the closer the number of speech sounds in the head is to one, (b) the greater the Consonantal Strength value of its onset, and (c) the more sharply the Consonantal Strength drops from the onset toward the Consonantal Strength of the following syllable nucleus.
    2. **Coda Law** A syllable coda is the more preferred: (a) the smaller the number of speech sounds in the coda, (b) the less the Consonantal Strength value of its offset, and (c) the more sharply the Consonantal Strength drops from the onset toward the Consonantal Strength of the preceding syllable nucleus. [Not quite the mirror image of the Head Law]
    3. **Nucleus Law** A nucleus is the more preferred: (a) the steadier its speech sound, and (b) the less the Consonantal Strength of its speech sound.
2. Preference laws for sequences of syllables
    1. **Weight Law** In stress accent languages an accented syllable is the more preferred, the closer its syllable weight is to two moras, and an unaccented syllable is the more preferred the closer its weight is to one mora. (The optimal stressed syllable is bimoric, the optimal unstressed syllable is unimoric.)
    2. **Law of Initials** Word-medial syllable heads are the more preferred, the less they differ from possible word-initial syllable heads of the language system.
    3. **Law of Finals** Word-medial syllable codas are the more preferred, the less they differ from possible word-initial syllable codas of the language system.
    4. **Strength Assimilation Law** If Consonantal Strength is assimilated in a syllable contact, the Consonantal Strength of the stronger speech sound decreases. [Generalisation of Murray's Progressive Assimilation Law.]
    5. **Contact Law** A syllable contact A$B is the more preferred, the less the Consonantal Strength of the offset A and the greater the Consonantal Strength of the onset B; more precisely - the greater the characteristic difference CS(B)-CS(A) between the Consonantal Strength of B and that of A.

To illustrate, using the Head Law: Vennemann adduces many examples and types of examples of language change, including consonant insertions if < 1, i.e. 0:

German: insertion of glottal stop before syllable-initial vowel:

ʔalt, ʔa.ʔor.ta, cha.ʔo.tisch, cha.ʔos

Italian: insertion of intervocalic glides, developing into obstruents:

Pau.lo > Pa.o.lo > {Pa.vo.lo, Pa.go.lo}

And consonant deletions if > 1:

OHG: loss of /h/: hwiz > wiz „white"

English: cf. *knee*, *gnome*, *gnat*

Optimality Theory (OT)

Optimality Theory has been one of the most productive methodologies for phonological description for a number of years. The approach would merit detailed explication on ist own in terms of non-monotonic reasoning, but the present context is not the place for this. Optimality Theory (OT, Prince & Smolensky 1993) has two main descriptive goals:

1. Assignment to some linguistic input of a structural description (representation) by selecting from a (possibly infinite) set of candidate structural descriptions the one which best satisfies a ranked set of universal constraints.
2. The establishment of different rankings of the universal constraints in order to account for the difference of representations in different languages.

The issue is frequently formulated procedurally as a search problem:

1. To select *competing representations* from a large search space of possible representations.
2. To select competing theories (constraint rankings) from a large search space of possible rankings.

Optimality Theory, like Vennemann's Preference Law approach, seeks the *optimal*, *best* representation; unlike Vennemann's approach it does not deal with the diachronic dimension (but see Löhken 1997). This means that there is no homogeneous definition of well-formedness; well-formedness is a function of the interaction of constraints. In a sense, this is not new. The traditional style of defining phonological well-formedness in generative phonology, going back 40 years, was by separate *redundancy rules* operating over lexical representations. The application of all redundancy rules yielded well-formed structures. However, no principles were defined for the assignment of lexical representations; these were conceived of as idiosyncrasies. The overall effect was oddly atomistic, in particular over domains which are quite well-defined, as the discussion of syllable structure in the context of automata and of Vennemann's Preference Theory shows. The atomistic effect is enhanced by use of the conventional style of argumentation: rather than dealing with whole systems, and attempting to capture their interactions, results are presented for selected isolated problems in fragments of widely different languages. This is not to say that the work is not based on careful intensive and extensive description; the opposite is true, in general.

But it is noteworthy that on the empirical side the bible of generative phonology (Chomsky & Halle 1998), for example, takes quite the opposite tack both in describing English phonology in intimate detail and in presenting a coherent theoretical framework.

It is also noteworthy that on the theoretical side, Vennemann starts with a general theory of syllable structure before following up with detailed analyses of interesting fragments: structural categories such as HEAD, BODY, NUCLEUS, CODA are coherently defined as a frame of reference for constraints.

Optimality Theory has little to say about the nature of the search problem, or about how to get beyond the implication of a simplistic „generate and test" strategy (the GEN function) of constraint satisfaction, though work has been done on this (Tesar 1995), and a number of partial implementations are available on the web. One of the favourite topics of OT descriptions is syllable boundary assignment (syllabification), because syllables provide domains for many phonological rules and, being a parsing problem, syllabification is a classic variety of search problem; see also Jurafsky & Martin (2000).

An example of tableau oriented OT argumentation is given in Table 1. the lowest number of violations is selected. In this case:

1. The columns contain the constraints (partially) ordered left-right in priority ranking order from strictest to least strict constraint.
2. The rows contain potentially infinitely many output candidates (freely generated).
3. The tableau is filled from left to right with evaluations by matching candidates with constraints: nothing is entered if the constraint is not violated, and an asterisk is entered (and the rest of the row is shaded) if the candidate violates the constraint.

4. The number of asterisks in a tableau cell is its value.
5. A simple algorithm for filling the tableau is given below.
6. Evaluation: Throw out the rows containing cells with exclamation marks and shadings (the shadings are actually redundant), and the remaining candidate or candidates win.

Table 1: An Optimality Theory tableau (Output2 wins).

Input: x1...xn	Constraint_1	Constraint_2	Constraint_3	Constraint_4
Output1	*!			
Output2			**	
Output3		*!		
. . .				

In fact the shading operation is not quite correct since only the loser rows are shaded; the winner is also shaded in the tableaux in the literature. The algorithm is also non-optimal and; because exhaustive for an infinite number of candidates it will clearly not terminate. The mnemonically named lower level procedures also require definition.

Partial OT tableau filling algorithm

```
DEFINE constraints{1,...,numberOfConstraints];
DEFINE candidates[1,...,numberOfCandidates];          // Too bad if the set is infinite.
i=0;
WHILE (i < numberOfConstraints) {
  j = 0;
  WHILE (j < numberOfCandidates) {
          IF (unshaded(cell[i,j] AND VIOLATE(cand[j],constraint[i]))
                  THEN addAsManyAsterisksAsViolations(cell[i,j]);
          j++}
  j=0;
  WHILE(j < numberOfCandidates){
          k=0;
          WHILE (k < numberOfCandidates){
                  IF (moreViolations(cell[i,j],cell[i,k])) {
                          THEN addExclamationPointIf NoneThereAlready(cell[i,j]);
                          l=i+1;
                          WHILE(l <= numberOfConstraints) {
                                  shade(cell[l,j]);
                                  l++;      }          }
                  ELSE IF moreViolations(cell[i,k],cell[i,j]]){
                          THEN addExclamationPointIfNoneThereAlready(cell[i,k]);
                          l=i+1;
                          WHILE(l<=numberOfConstraints){
                                  shade(cell[i,k]);
                                  l++;      }
                  k++;      }
          i++;      }
```

The following example of syllabification in German is taken from Löhken (1995), p. 17:

Constraints:

ONSET	syllables have an onset
FILL	structural positions are filled with underlying segments
NOCODA	syllables do not have a coda

Tableau (candidate 2 wins):

Input: teatr	ONSET	FILL	NOCODA
te.a.tr	*!		*
te.a.tr		*	*
te._a.t.r		**!	

Tesar's study (1995) investigates algorithms for feasible calculation of optimal candidates and for learning rankings, However, a different strategy would be more in keeping with current developments in computational linguistics: Rather than (conceptually) a dubious *generate and test* approach to providing candidates, and stipulation of constraints in a ranking order, it would be more sensible to define operations over the constraints themselves and the input. The result of the operations would be the successful candidate. As long as there is a finite set of constraints, and operations which terminate, this strategy will also terminate. In fact, some attempts to define *a priori* relations between input and output have been made in the literature (Löhken 1997, p. 13), and Karttunen adduces a number of studies, in addition to his own, which show that the OT GEN function and constraints can be modelled by regular relations. However, the Penn-Thomason Machine, PTM, approach, and default unification suggest themselves as possible avenues to explore.

Karttunen's discussion of OT

Playing on the title of a paper by Smolensky „On the proper treatment of connectionism", which in turn calques the title of a classic paper in formal semantics by Montague „On the proper treatment of quantification in English", Karttunen (1998) discusses „The Proper Treatment of Optimality in Computational Phonology", criticises the sub-optimal tableau evaluation method, and shows that „the computation of the most optimal surface realizations of any input string can be carried out entirely within a finite-state calculus"; see also Jurafsky & Martin (2000). He points out just how sub-optimal the optimality theoretic method is:

> Because GEN over- and underparses with wild abandon, it produces a large number of output candidates even for very short inputs. For example, applying GEN to the string *a* yields a relation with 14 strings on the output side ... The number of output candidates for *abracadabra* is nearly 1.7 million, although the network representing the mapping has only 193 states. It is evident that working with finite state tools has a significant advantage over manual tableau methods.

The alternative approach - operations over constraints, and definition of input/output relations - is exactly the approach taken by Karttunen, who introduces a new operation of *lenient composition*, related to *priority union*. It is also related to and default unification, an operation which has cropped up a number of times in discussion. Karttunen summarises the properties of his approach:

- No marking, sorting, or counting of constraint violations.
- Application of optimality constraints is done within the finite state calculus.
- A system of optimality constraints can be merged into a single constraint network

Effectively, the highly procedural features of OT are replaced by a declarative functional concept, cascades of finite state transducers, with a single rule, namely the state transition.

Quo vadimus?

To look for a unified approach to defaults (or to their avoidance) in phonology is surely not as hard as finding a unified theory of electromagnetic and gravitational waves. But, as my survey has shown, there is such a wide range of uses of default-like constructs, and notational devices for expressing them that the twin tasks of finding a unified approach and persuading phonologists, phoneticians and others to accept it is by no means trivial. A possible path forward will no doubt involve the following set of heuristic strategies:

1. Systematically identify basic explicanda such as feature markedness, feature-changing rules, context specificity and elsewhere conditions, overriding of preference laws, constraints, Bayesian results by interaction of modules.

2. Systematically identify, interrelate and extend basic formal devices such as default unification, priority union, ANY-loop automata, default automata, default inheritance.
3. Aim for description of whole language systems rather than picking the cherries out of the cake and describing juicy fragments; if this is not done, then there is hardly any hope of addressing the inter-module overriding issue.
4. Aim for clean implementation of results for theory testing and in human language technology applications.
5. Be patient with the calls from engineers to be more compact, more efficient, faster, more accurate, use more statistics.
6. Convince (and I mean *convince*!) engineering colleagues that a powerful future systems require structure as well as statistics.
7. Structure and statistics are orthogonal, not in competition: use statistics to augment and induce structure, not to replace it.
8. Evaluate implemented models on corpora with accepted statistical methods.

But the main task will be to define a *preference space* of syntagmatic, paradigmatic and mapping default-override relations, and to find a family of related mechanisms to express these.

References

Bird, Steven & Ewan Klein
1989 Phonological Events. Journal of Linguistics 26:33-56.

Bleiching, Doris, Guido Drexel & Dafydd Gibbon
1996 Ein Synkretismusmodell für die deutsche Morphologie. In: D. Gibbon, ed., Natural Language Processing and Speech Technology. Results of the 3rd KONVENS Conference, Bielefeld October 1996. Berlin: Mouton de Gruyter. 237-248.

Bouma, Gosse
1992 Feature structures and nonmonotonicity. Computational Linguistics 18, 183-203.

Coleman, John
1998 Phonological Representations: Their names, forms and powers. Cambridge: Cambridge University Press.

Carson-Berndsen, Julie
1998 Time Map Phonology: Finite State Methods and Event Logics in Speech Recognition. Kluwer Academic Press: Dordrecht.

Chomsky, Noam & Morris Halle
1968 The Sound Pattern of English. New York: Harper & Row.

Devlin, Keith
1991 Logic and information. Cambridge: Cambridge University Press.

Gibbon, Dafydd
1976 Perspectives of Intonation Analysis. Bern: Lang.

Gibbon, Dafydd
1985 Prosodic parsing in English and Akan. Paper given at the 21st International Conference on Contrastive Linguistics, Blazejewko, Poland.

Gibbon, Dafydd

1987 Finite state processing of tone systems. In: EACL 3 Proceedings, Copenhagen. 291-297.

Gibbon, Dafydd
1990 Prosodic Association by Template Inheritance. In: Gazdar, G. & Daelemans, W., eds., Proc. Int. *Workshop on Inheritance in Natural Language Processing*. Tilburg.

Gibbon, Dafydd
1991 DDATR: un logiciel de traitement d"héritage par défaut pour la modélisation lexicale. In: Cahiers Ivoiriens de Recherche Linguistique (CIRL) 27, 5-59.

Gibbon, Dafydd
1992a ILEX: A linguistic approach to computational lexica. In: U. Klenk, ed., Computatio linguae. Aufsätze zur algorithmischen und quantitativen Analyse der Sprache, Beihefte der Zeitschrift für Dialektologie und Linguistik. S. 32-53.

Gibbon, Dafydd
1992b Prosody, Time Types and Linguistic Design Factors in Spoken Language System Architectures. In: G. Görz (ed.), KONVENS 92 1. Konferenz „Verarbeitung natürlicher Sprache". Berlin: Springer-Verlag, 90-99.

Gibbon, Dafydd
2000 Computational lexicography. In: van Eynde, F. & D. Gibbon: Lexicon Development for Speech and Language Processing. Dordrecht: Kluwer Academic Press.

Jurafsky, Daniel & James H. Martin
2000 Speech and Language Processing. An Introduction to Natural LanguageProcessing, Computational Linguistics and Speech Recognition. Upper Saddle River, NJ: Prentive Hall.

Hawkins, John A.
1994 A performance theory of order and constituency. Cambridge: Cambridge University Press.

Hirst, Daniel & Albert Di Cristo, eds.
1998 Intonation Systems: A Survey of Twenty Languages. Cambridge: Cambridge University Press.

Johnson, C.D.
1972 Formal Aspects of Phonological Description. The Hague: Mouton.

Kaplan, Ron M. & Martin Kay
1994 Regular models of phonological rule systems. Computational Linguistics 20(3), 331-78.

Karttunen, Lauri
1998 On the proper treatment of Optimality Theory in Computational Phonology. In: SIGPHON 1997.

Kay, Martin
1983 When Meta-Rules are not Meta-Rules. In: Karen Sparck Jones & Yorick Wilks

(eds.) Automatic Natural Language Parsing. 94-116.

Keating, Patricia A.
1988 „Underspecification in phonetics“, Phonology 5.2 , 275-292

Koskenniemi, Kimmo
1983 Two-level morphology. Ph.D. thesis, U Helsinki.

Ladd, D. Robert
1980 The structure of intonational meaning. Bloomington: Indiana University Press.

Ladd, D. Robert
1996 Intonational Phonology. Cambridge: Cambridge University Press.

Liberman, Mark Y. & Alan Prince
1977 On stress and linguistic rhythm. Linguistic Inquiry 8, 249-336.

Löhken, Sylvia
1997 Deutsche Wortprosodie: Abschwächungs- und Tilgungsvorgänge. Tübingen: Stauffenburg Verlag.

Marslen-Wilson, W. D.
1987 Functional parallelism in spoken word-recognition. In: U.H. Frauenfelder & L. Komisarjevsky Tyler (eds.), Spoken Word Recognition. Amsterdam: Elsevier Science Publishers B.V.: 71-102.

Penn, Gerald & Richmond Thomason
1994 Default finite state machines and finite state morphophonology. SIGPHON. Ms.

Prince, Alan & Paul Smolensky
1993 Optimality Theory: Constraint interaction in generative grammar. Technical Report CU-CS-696-93, Department of Computer Science, University of Colorado at Boulder, and Technical Report TR-2, Rutgers Center for Cognitive Science, Rutgers University, New Brunswick, NJ.

Reinhard, Sabine & Dafydd Gibbon
1991 Prosodic Inheritance and Morphological Generalisations. Proceedings, ACL European Chapter Conference, Berlin, April 1991.

Selkirk, Elizabeth O.
1984 Phonology and Syntax: the Relation between Sound and Structure. Cambridge, MA: MIT Press.

Tesar, Bruce
1995 Computational Optimality Theory. Ph.D. dissertation, University of Colorado.

Trubetzkoy, N.
1939 Grundzüge der Phonologie. Prague: Cercle Linguistique de Prague.

Vennemann, Theo
1988 Preference Laws for Syllable Structure and the Explanation of Sound Change. Berlin: Mouton de Gruyter.

Wells, John. C.
1989 Computer-coded phonemic notation of individual languages of the European Community. Journal of the International Phonetic Association 19(1): 31-54.