Typology of African Prosodic Systems (TAPS)

Workshop at Universität Bielefeld, Germany, 18−20 May 2001

Computational phonology and the typology of West African Tone Systems

Dafydd Gibbon, University of Bielefeld

OVERLONG ABSTRACT or PRELIMINARY DRAFT

1. Computational contributions to the adequacy of linguistic descriptions

This paper is both methodological and descriptive. It is methodological in that it looks at possible contributions of the methodologies of computational phonology to the typology of tone systems and is complementary to papers by Urua, Gut, Gibbon, Adouakou. It is descriptive, in that it considers direct linguistic observation of data, model−directed phonetic measurements of tone, computational properties of these models, and *explicanda* for tone theories from the phonological literature.

Many structural properties of register tone systems are quite well understood, and a number of general principles have been formulated which characterise the constraints on possible tone systems. Among others, these include Leben's Obligatory Contour Principle (OCP), Goldsmith's well−formedness principles for tone−syllable association, which I will refer to as Association Principles (AP), and Clements' right−branching metrical model of tonotactics. There are also more specific preferences, which will be listed below. However, there are still some major theoretical gaps. For example:

1. *Observational adequacy*: as yet, there is not a full model of the mapping between symbolic phonetic and quantitative phonetic descriptions. Individual studies have been made by Liberman & al. (various studies); Connell & Ladd; Bird; Ahoua; Ladd; Gibbon, Urua & Gut, and others.
2. *Descriptive adequacy*: there is no coherent underlying tonal "grammar" or set of constitutive constraints with known formal properties. This weakness is shared by Optimality Theory (OT), which has no general theory of underlying structure. Invididual statements of constraints on tone patterning are to be found in numerous descriptive grammars, though it has been suggested that finite state models (Gibbon) are adequate for this purpose.
3. *Explanatory adequacy*: there are few known general principles from which the similarities and differences between specific tone systems can be deduced. Again, this weakness is shared with Optimality Theory approaches, which have no general principles from which specific constraint orderings can be derived. Exceptions can be found, however, in a number of specific areas of tonogenesis and comparative studies of specific languages and dialects.

In the context of explanatory adequacy, a central question which is often posed is that of the autonomy of tone systems. There are a number of items of evidence for relative autonomy, such as the independence of tone sandhi rules from particular lexico−syntactic sequences, the robustness of tone in language change leading to floating low tones in explanations of automatic downstep and possibly other forms of lowering. Formal questions about the interdependence of tone and other categories include the following:

- Are there well−motivated preferences for associating complexity of syllabic structure with complexity of tonal or accentual structure?
- Are there well−motivated preferences for typical nominal structures as opposed to typical verbal (or other) structures?
- Are there well−motivated preferences for realisational dependencies (segmental, phrasal domain, level sequencing)?

In this contribution, I would like to point to a number of strategies which have been developed over the past two decades in computational linguistics and speech technology, and suggest that these strategies may be useful indicators for theoretically more interesting empirically founded descriptive and explanatory principles in linguistic description, starting from a basic premise that a description of tonal

systems necessarily has the just following three components:
1. Inventory of lexical tonal categories (Component 1);
2. Tonotactics of tone sequences in lexical and phrasal contexts (Component 2);
3. Realisation mapping of lexical tones in context as allotones, i.e. tonal sandhi (Component 3).
Relations between language systems are regularly defined in terms of markedness, implicational universals, and correspondence rules. In this first approximation I will use the term "preference"", following a long tradition in natural phonology, and I will explicate preferences formally in terms of defaults and overrides. Each of the three components listed here will be characterised in terms of preferences.

2. Tone system preferences

Lexical tone inventory and category mapping preferences (Component 1)
African tone inventories (maybe some others, too) show lexical inventory component preferences such as the following, which pertain to the size of the inventory, the categories of tones in the inventory, and a preference relation among tones:

- a preference for a tone inventory size in the following order: 2 < 3 < 4 < ... ;
- a preference for level target tones over contour tones;
- a preference for low tones over high tones as default tones;
- a preference for reducing contour tones to sequences of level tones.

An inventory is finite, though the contexts to which its elements may be conjoined is not necessarily finite. The finite lexical inventory can conventionally be represented in terms of a microstructure (a feature matrix, possibly hierarchical as in feature geometry models); a mesostructure (generalisations over the microstructure in terms of redundancy rules or as an implication or inheritance hierarchy); and a macrostructure (organisation of simplex and complex lexical items into a list, tree, etc.). The other components are grounded in the lexical inventory component; I will not have much to say about this here, though some studies on the formal modelling of tone in the lexicon are available.

Tonotactic preferences (C2)
The constitution of structures is conventionally formulated in terms of a grammar, of which there are many types and flavours. For West African tone systems, preferences like the following can be listed:

- a preference for a metrical structure which is essentially right branching;
- a preference for lexical category dependent tonotactics, e.g. freely combinable tones on Nouns, restricted tones on Verbs (Baule−Anyi); pitch accent like constraints (Tem) rather than entirely free combination.

It is a well known result in the theory of formal languages that right−branching structures can easily be modelled by linear devices in the technical sense of the term, i.e. regular grammars (Type 3 formal grammars) and finite state automata (FSAs). In previous work I have shown that tone patterning can indeed be modelled by FSAs. So in the mathematical sense, the grammar of tone is indeed linear, meaning "head or tail recursive (i.e. iterative), but not arbitrarily recursive". The established but informal terminology of "linear vs. non−linear phonology" has a different meaning, for which I will use the term "multilinear" in order to avoid confusion. Aspects of finite state tone modelling first appeared in Gibbon (1987) and have been discussed in a number of conference presentations since then.

Mapping preferences (C3)
The realisation of underlying structures constituted by the tones of lexical items, in the context of other tones and of segmental and phrasal categories, can also be characterised in terms of general preferences:

- a preference for sandhi mapping to be dependent on phrasal, not lexical domains (domain of upsweep, verb subcategorisation in Kwa);
- a preference for tones to be realisationally robust in comparison with syllabic structure, particularly vowels;
- a preference for no, or minor segmental effects on tone realisation (depressor consonants, intrinsic

vowel pitch);

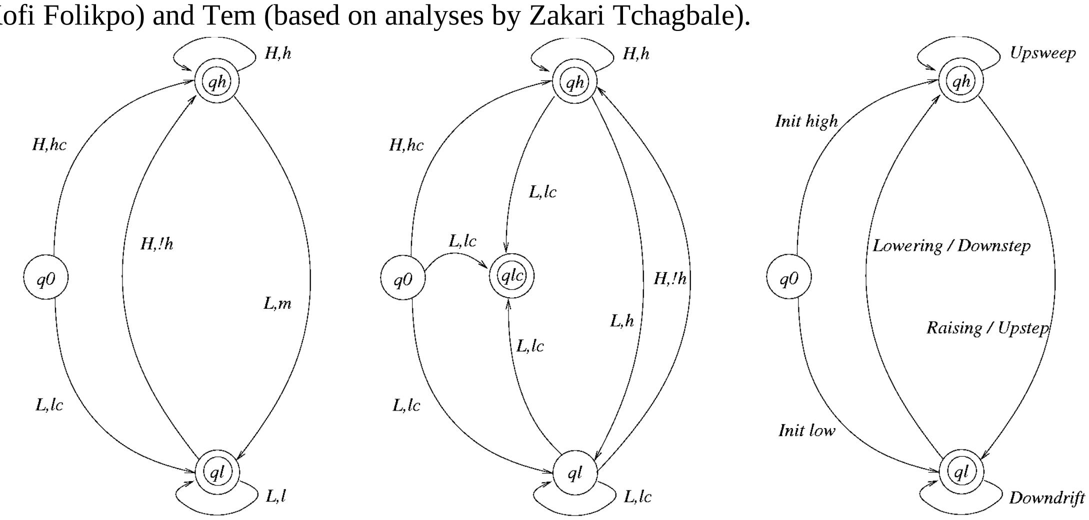
- a preference for terraced tone levels (and 2 lexical tones) rather than discrete tone levels (and 3 or more lexical tones).

From the computational point of view, the mapping is from one level of lexically determined linear structure (in the context of linear configurations of other lexical categories) to another level of phonetic linear structure. This multilinear mapping is relatively simply modelled by a variant of the FSA which operates not with single symbols but with pairs (or larger tuples) of symbols.

The automaton type concerned is the *finite state transducer* (FST), which has been used in computational phonology and prosody for around three decades, and for two decades has been the accepted standard formalism for phonological and prosodic modelling; this also applies to speech recognition technology, where the standard Hidden Markov Models (HMMs) are probabilistic variants of FSTs, and are incidentally also known as a variety of linear system. This approach to the modelling of tone mapping was formulated in some detail for Tem (Gur) and Baule (Tano/Kwa) by Gibbon (1987). The status of FSAs and FSTs have become standard tools to such an extent that "Finite State Technologies" has become a mainstream paradigm in many areas of language and speech processing, and also in many other areas of technology, including bioinformatic processing.

3. Tone system preferences

The realisation models add constraints by providing contexts for different allotone mappings at different points in the topology of the finite state model. Models in the form of finite state transition networks (a formal visualisation of FSTs) are shown in the Figures for Ewe (based on an analysis of H/L sequences by Kofi Folikpo) and Tem (based on analyses by Zakari Tchagbale).



The Ewe automaton shows the basic form of a *start state* and an oscillation between two states, one for each tone, in this case a *high state* and a *low state*. The automaton abstracts away from many aspects of the tone system, but it clearly represents the main tone sandhi effects in Ewe, i.e. Relatively consistent startup tones (high and low), raising of the first low after a high to mid, and downstepping of the first high after a low.

This automaton type can be generalised to other tone systems. Transitions between these three states suffice for defining the main contexts for allotone mappings of the types attested in the literature:

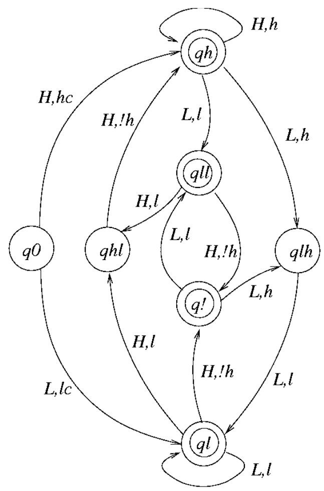
- *start−high*: startup effects, in particular a putative constant high and constant low target tone;
- *high−high*: high *demiterrace*, in terms of which level or upsweep high sequences are defined;
- *high−low*: transition from high to low, in which the first low of a sequence may be raised to mid or high.
- *low−low*: low *demiterrace*, in terms of which low constant or low downdrifting tones are defined; *low−high*: the context in which automatic downstep is defined (or, if floating low tones are included and may be taken to have the properties of overt tones, non−automatic downstep too);

- full *terrace* is defined as a complete cycle between the two tonal states, i.e. from high to low and back to high, or from low to high and back to low.

It will be a pleasant puzzle for the gentle reader to solve the question of how to model a super−high tone at the end of a sequence of high tones in this type of model.

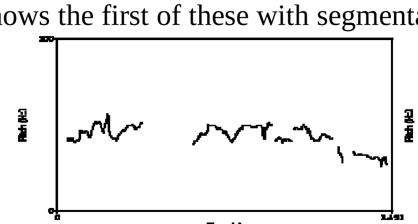
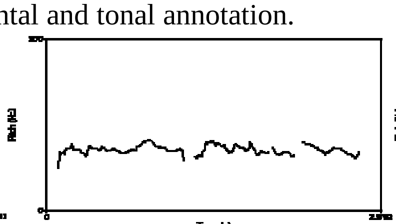
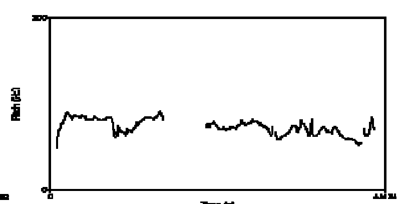
A generalised finite state oscillator is shown in the third Figure. The terminology which is familiar from the literature is formulated as labels on the relevant transitions.

The known facts for Baule are more complex. The relevant contexts are longer: it is not only adjacent contexts which are relevant, but longer sequences, which may be modelled by an additional oscillation for the more complex contexts; this is modelled by the inner loop in the Baule figure.

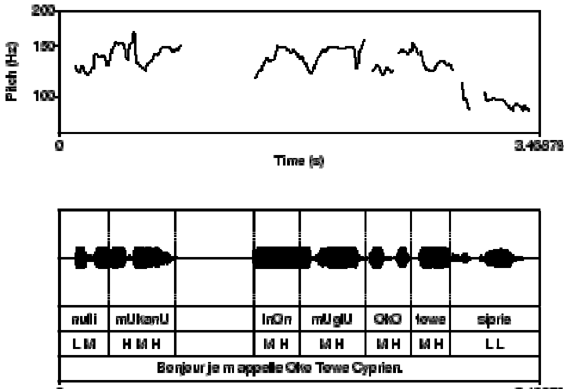


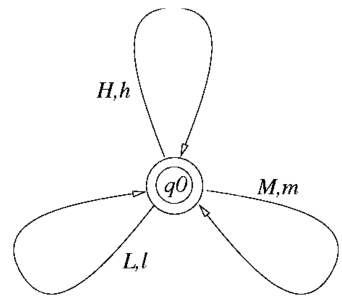
It must be noted that this class of models, as it stands, does not account explicitly for dependencies between tones and segment types, syllable types, or lexical and phrasal domains. The relation to phrasal domains in the sense of Ahoua and Leben can easily be made explicit, however: the start state coincides with the start of a phrasal domain, and an end state (the latter are conventionally denoted with two circles), of which there are more than one, coincides with the end of a phrasal domain.

So far we have considered two−tone systems. What of systems with more than two tones? An example of such a system is Ega, which Ahoua, Connell and I are examining in our current project "Ega: a documentation model for an endangered Ivorian language". Ega has been analysed as a three−tone language. Following the − possibly perceptually motivated − preference for discrete level tone systems if there are more than three tones, high demi−terraces in Ega are on the same pitch, without automatic downstep, and low demi−terraces are also on the same low pitch. In several examples of formal speech from different speakers which have been recorded and documented, the two demi−terraces are separated by a minor third interval, that ubiquitous interval which has been repeatedly reported and speculated about in many accounts of pitch intervals in intonation and tonal prosody. There is a clear final lowering effect, however. Three consecutive utterances are shown in the following three figures; the next figure shows the first of these with segmental and tonal annotation.

But we have already noted that Ega has a discrete level tone system, therefore the motivation for the automatic downstep transition is absent. And so far, we have no evidence for other contextual effects (though this is not to say that other effects may be noted in the future). So if all tones behave alike, essentially, and there are no constraints, then the model simply collapses into a "freewheeling" model such as the one shown in the next figure(a similar model, but with two loops, could have been postulated for an entirely constraint−free two−tone system). If more contexts for the tone realisation mapping are found, then this extremely simple model will need to be expanded again, with the introduction of further states in order to provide more transitions as contexts for the constraints.





4. On the contribution of computation: grammars, not trees or lattices

So far I have not mentioned computers, processing, processes, procedures, algorithms, rules and the like. Neither have I mentioned representations. I have discussed grammars, however, implying that a grammar is a set of generalised representations from which specific representations can be inferred by very general principles of deduction. And this is the particular point of progress in the modelling of tone, to which I would like to lay claim. Previous descriptions have had, in general, the following characteristics:

1. concentration on representations of specific sequences, for example as autosegmental diagrammes, or as metrical trees;
2. formulation of generalisations between specific representations as separate rules (e.g. downstep, assimilation);
3. ignoring of the holistic properties of the representations into account;
4. ignoring of the formal properties of the rules (are they processable, are they learnable);
5. ignoring the explanatory requirement of putting the separate rules into some kind of structural context to the others.

This drastic critique is perhaps not as drastic as it sounds. For instance, Clements' explication of the metrical structure of tone is in terms of (essentially) right branching trees. What is missing is a grammar which defines these trees. It turns out that these trees are very simply defined by means of a linear (regular, Type 3) grammar, equivalently by a finite state automaton. And the kind of FSA which generates them is − not coincidentally, of course − the kind which I have described above as models for the tonal realisation mapping constraints.

The contribution of computation lies, therefore, not in the use of computers, though they are extremely helpful in enforcing rigour in developing models for theoretical approaches, nor even in discussion of details of processing. The contribution of computation is to provide a clear language for asking appropriate questions in order to push the field forward, and for clarifying distinctions which may not always have been clear, between representations, rules and algorithms in such a way as to result in a fully explicit and computationally testable model for a theory. An example in which a computational approach is helpful is in the fundamental distinction which is widely adhered to in the computationally oriented sciences between *declarative information* and *procedural information*:

1. *Declarative information*: well defined structures and generalisations over structures, such as lists and trees of various shapes, tables, networks with various properties, linked by a minimum of procedural rules of composition and derivation.
2. *Procedural information*: rules of inference, either general operations such as modus ponens, unification, specific operations such as logical or algebraic substitution rules, as in typical linguistic

rules.

With these two orthogonal concepts it is straightforward to systematise the debate on constraint−based vs. derivational approaches.

The relation between the declarative and procedural components of a formal theory can be stated in complementary ways:

1. by derivation: a structure is derived by rules of inference from structural axioms;
2. by filtering: a rich set of structures is restricted by intersecting constraints.

A theory such as OT combines both strategies: constraints act as filters, but are ordered like derivational rules, as Karttunen has convincingly (and humorously) shown.

Finally, it is also useful to note that none of what has been discussed so far pertains to the notion of algorithm, a term which is used rather loosely in linguistics. An algorithm is a procedure for calculating the result of a specific problem which will terminate in a finite number of steps. There are many kinds of procedure which do not terminate − a trivial case is when your PC crashes, this is for instance its terminal state. The procedure ends in an infinite loop because the software developers have not correctly implemented the appropriate algorithms.

5. Conclusion: computation, linguistic adequacy and typology

In summary, I would like to make the following points:

1. Computational models contribute to observational adequacy by providing formal models on which to base quantitative studies. In this overview I have not touched on studies of this kind, but have kept the approach complementary to ongoing work by Urua, Gut and myself. This work continues the long line of research by scholars such as the following: Fujisaki, ’t Hart, Cohen & Collier, Pierrehumbert, Liberman on intonation, and Liberman et multi alii; Connell & Ladd; Laniran; Ahoua; Gibbon, Urua, Gut on tone.
2. Computational models contribute to descriptive adequacy by providing frameworks for grammars which express lingusitically significant generalisations, rather than listing representations (however interesting and complex, and rules (however intricate and however many). In particular, the FST model directly expresses the organisational principle underlying metrical analysis: *rhythm*, i.e. the (temporally regular) oscillation between two states of the same empirical parameter. Temporal regularity is not the main issue in the context of tone, of course, but the concept of metre, or rhythm, as oscillation, rule−governed alternation, is central. In the FST model, this alternation is modelled by iterative loops.
3. Computational models contribute to explanatory adequacy by providing a clear basis for a range of questions connected with explanatory adequacy, such as:
   a) What are the simplicity and complexity measures for tone systems, and is there an upper bound on complexity with checks and balances to distribute complexity between the tonal inventory, the tonotactics and the tone realisation mapping?
   b) What are the general principles behind the different kinds of tonogenesis?

From the computational point of view, if it has been established, and I think it has, that finite state devices are adequate for modelling tonal systems, then these questions may be reformulated slightly in terms of the differences between finite state models with different network topologies (tonotactics) and different realisational vocabularies.

So what are the prospects for the typology of tone systems? I suggest that the use of simple, working models in which tone realisation mapping constraints are put into a coherent, connected overall context, provides a solid basis for expressing and visualising the different topologies of tone systems. As stated at the outset, the present contribution is a methodological one, but founded on a particular type of empirical modelling: automatic pattern matching of well−defined and fairly simple models with quantitative measurements of data. It would go too far to discuss the principles of comparing the different network structures on which typological studies may be based. However, notations exist in which comparisons of this kind may be made relatively easily. One example is shown in the appendix, an working, i.e. operational logical model for tone FSTs which can be used to test the derivation of phonetic patterns from underlying lexical patterns. For example, the following results were obtained with an implementation of this logic:

```
Tone: <H H H L L L H H H L L L> = hc h/ h! ^l l‘  l^ !h h/ h! ^l l‘  l‘  l% .
Baule:<H H H L L L H H H L L L> = hc h h h l l l !h h h l l .

Tem:  <H H H L L L H H H L L L> = hc h h h lc lc !h h h h lc lc .
```

Note that although the underlying patterns are the same ,the outputs on the right hand side are different (for details see the Appendix).

Still, most of the questions posed in the course of this contribution are far from having been answered in the present contribution. But I suggest that at least we have a framework now for putting the questions has been developed, and some of the questions have been answered. And for those interested in technological applications of African languages, we have provided an essential intellectual tool taken from the Finite State Technology paradigm.

[BIBLIOGRAPHY TO BE ADDED]

Appendix: Formalisation of tone FST models

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% tone.dtr
% D.Gibbon, 2000.06.18
% Register tone automaton
% for West African tone languages
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Encoding
% Lexical/underlying tones: upper case H, L
% Phonetic/surface tones:   lower case h, l with diacritic
% lc = low constant               l^ = regressively raised low
% ^l = progressively raised low    l\ = downdrift low
% %l = final lowering              hc = high constant
% h! = regressively lowered high  !h = progressively lowered high
% h/ = upsweep high                %h = final raising
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Declarations
# atom H L .
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Model 1: Generalised model
Tone:
       <L> == 'lc' Tone_l:<>
       <H> == 'hc' Tone_h:<>
       <> ==.
Tone_l:
       <L H> == 'l^' <H>
       <L> == 'l' ' <>
       <H> == '!h' Tone_h:<>
       <> == '%l'.
Tone_h:
       <H L> == 'h!' <L>
       <H> == 'h/' <>
       <L> == '^l' Tone_l:<>
       <> == '%l'.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Model 2: Tem
Tem:
        <H>  == hc Tem_q1:<>
        <L>  == lc Tem_q2:<>
        <>   == .
Tem_q1:
        <H>  == h <>
        <'*L'> == lc
        <L>  == h Tem_q2:<>
        <>   == Tem.
Tem_q2:
        <H>  == '!h' Tem_q1:<>
        <L> == lc <>
        <>   == Tem.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Model 3: Baule
Baule:
        <H>   == hc Baule_q1:<>
        <L>   == lc Baule_q2:<>
        <>    ==.
Baule_q1:
        <L L> == h l Baule_q2:<>
        <L>   == l Baule_q2:<>
        <H>   == h <>
        <>    ==.
Baule_q2:
        <H H> == l '!h' Baule_q1:<>
        <H>   == '!h' Baule_q1:<>
        <L>   == l <>
        <>    == .
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```