Computational phonology and the typology of West African Tone Systems

Dafydd Gibbon, University of Bielefeld
gibbon@spectrum.uni-bielefeld.de

1. Computational methodology and prosodic description

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 at this conference by Akinlabi, Connell, Ladd, 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 descriptive models, and *explicanda* for tone theories from the phonological literature. The paper is also traditional, in that it takes up a long tradition of finite state (FS) modelling in prosody (intonation, accent and tone), from Fujisaki[1] in the 1960s to the IPO group in the 1970s, Pierrehumbert, Liberman and others in the 1980s and 1990s, myself with various contributions in the 1980s and 1990s, and the extensive work in finite state phonology which is summarised and continued in Carson-Berndsen (1998). The paper breaks new ground, however, in taking FS modelling beyond the area of observational adequacy to the issue of the descriptive adequacy of prosodic grammars, and the explanatory adequacy of comparative, typological descriptions of prosodic systems.

2. Formal aspects of prosodic typology: tone system parameters

In a very useful overview article, Schuh (1978) systematically discussed a range of mechanisms used in descriptions of lexical tone. A revised overview (note: not of tone systems, but of descriptions of tone systems) might include the following:

(1) *formalisms for tone*: representations of specific units, generalisations (rules), algorithms;

(2) *tone inventories*: 2-tone vs. >2-tone systems, features (African?) vs. units (Asian?), height vs. register; segmental vs. suprasegmental feature;

(3) *tone mapping* (realisation) operations: intrasegmental (assimilation, dissimilation, e.g. raising), intersegmental (spreading/dumping, displacement), downtrends (terraced vs. discrete level, terrace downdrift, demi-terrace downdrift, automatic and non-automatic downstep), uptrends (upglide, upsweep, upstep);

(4) *tone mapping domains*: phonological (neighbouring syllables, words), morphosyntactic (simplex and complex word, noun phrase, larger domains);

(5) *tone mapping triggers*: (near-)neighbouring tone, licensing vs. blocking consonant, morphosyntactic category;

(6) *semiotic functions of tone* and their interactions: lexical, morphosyntactic (inflexion, derivation, compounding), intonational (utterance type, structure;

---

[1] References have been restricted to those most immediately relevant to the argumentation.

focus, emphasis; textual, discoursal).

The present contribution addresses points (1), in respect of a finite state formalism for generalisations about tone, (2), in respect of 2-tone and >2-tone systems, and (3), in respect of intrasegmental mapping operations such as tone assimilation (raising, lowering) and downstep. The contribution starts from the claim that generalisations about tone (tone rules) are usually formulated as *filters*, i.e. statements which take some kind of input from somewhere and produce a well-defined output. In this respect, traditional segmental and autosegmental tonological rules are rather like Optimality Theoretic constraints; the generation of basic, well-formed structures is left as an open issue. The input to rules is symbolised by means of abstract data structures such as (toneme) *strings*, (tonal feature) *matrices*, (metrical) *trees*, (autosegmental) *lattices*. But, in general, no basic, coherently recognisable *grammar* of tone is usually given. This contribution seeks to substantiate the following claims:

- Tone has metrical - sequentially alternating - structure.
- The appropriate formalisation of metrical structure is a finite state grammar with a specific "oscillatory" or "loop" structure.
- The appropriate formulation of the grammar of tone is in terms of finite state grammars with this structure.
- Finite state grammars of this kind provide a useful foundation for typological comparisons of tone systems.

Some aspects of this claim have been anticipated in the finite state prosody and phonology literature, of course, as noted above. But the details are new, and the application to typology is new.

3. Adequacy

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:

- *Observational adequacy*: as yet, there is not a full model of the mapping between symbolic phonetic and quantitative phonetic descriptions. Individual experimental studies have been made by many scholars over the past four decades, and computational modelling and pattern-matching methods have been introduced in order to grapple with the size of the search space for this mapping (Liberman & al. in various studies; Bird; Ahoua; Connell & Ladd; Gibbon, Urua & Gut, and others).
- *Descriptive adequacy*: as yet, no reasonably sound or complete tonal "grammar" or set of constitutive constraints with known formal properties has been postulated. This property is shared by Optimality Theory (OT), which in most variants also has no general theory of underlying structure. Individual statements

of constraints on tone patterning are to be found in numerous descriptive grammars. It is suggested in the present contribution that a general theory is necessary, and that finite state models are adequate for this purpose.

- *Explanatory adequacy*: there are few known general principles from which the similarities and differences between specific tone systems can be deduced. Again, this property is shared with Optimality Theory approaches, which in most variants 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 in comparative studies of specific languages and dialects, where a number of basic principles have been formulated.

4. Generalisations: preference theories and default models

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 kinds of evidence for relative autonomy, such as the independence of tone sandhi rules from particular lexico-syntactic sequences, the robustness of tone in both synchronic paradigm formation and in language change, leading to the postulation of floating low tones in explanations of automatic and non-automatic downstep. Formal questions about the interdependence of tone and other categories include the following:

- Are there well-motivated preferences for associating complexity (or simplicity) of syllabic structure with simplicity (or 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, tone mapping sequences)?

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. The work reported in the present contribution is located within computational phonology, specifically computational prosody, with the long term goal of formulating the grammar of prosody in a computationally explicit fashion, for instance in terms of the theses above, and of explicating preferences (markedness, implications) in terms of default-override inference systems. A number of kinds of preferences would need to be addressed in the present context, though an explicit treatment presupposes the kind of foundational tonal grammar proposed in the present contribution, and cannot be dealt with in detail here. The relevant preference types are preferences for lexical tone, for tonotactics, and for tone mapping.

1) Lexical tone inventory preferences: African tone inventories (maybe some others, too) may show lexical inventory 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: $2 < 3 < 4 < \dots$;
- 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 complex units into 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 type/default inheritance hierarchy); and a macrostructure (organisation of simplex and complex lexical items into a list, tree, etc.). The other components of a grammar are grounded in the lexical inventory; I have nothing to say about this here (but cf. Gibbon & Ahoua 1991 on the modelling of tone in the lexicon using a default inheritance formalism).

2) Tonotactic preferences: The construction of complex units is conventionally formulated in terms of grammars, 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 (Anyi-Baule); pitch accent like constraints (Tem) rather than entirely free combination.

I will not address the second issue here. But 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 "linear"), 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 strict, technical sense, the grammar of tone is indeed linear. The patterns are indeed recursive, since they may be indefinitely long, but they are "head recursive" or "tail recursive", i.e. iterative, and not arbitrarily recursive. The established but informal terminology of "linear vs. non-linear phonology" has a different meaning, which can be confusing in computational contexts. For this I will use the term "multilinear" in order to avoid confusion.

3) Tone mapping preferences: Underlying structures are constituted by the tones of lexical items and realised in the context of other tones and of segmental and phrasal categories. In tone mapping, too, the patterns can 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);
- a preference for tones to be realisationally robust in comparison with syllabic structure (floating tones);
- a preference for no, or minor segmental effects on tone realisation (depressor consonants);
- a preference for consonant rather than vowel influence on tone realisation;
- 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 can also be modelled by a variant of the FSA which operates not with single symbols but with pairs (or larger tuples) of symbols (this point will be taken up below).

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 an accepted standard formalism for phonological and prosodic modelling. The same also applies to speech recognition technology, where the standard Hidden Markov Models (HMMs) are probabilistic variants of FSTs. The FST approach to the modelling of tone mapping was formulated in some detail for Tem (Gur) and Baule (Tano/Kwa) by Gibbon (1987). 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), with a number of readily available FS toolkits for experimentation and modelling.

5. Explanatory adequacy revisited: Comparing FS models of tone systems

Even if lexical tones may occur in arbitrary orderings, the tone mapping realisation models add constraints on possible sequences of surface tones. The topology of an FS network model defines a number of contexts for different allotone mappings at different points in the model. Models in the form of finite state transition networks (a formal visualisation of FSTs) are shown in the networks for Ewe (based on an analysis of H and L sequences by Kofi Folikpo, omitting the Ewe M tone) and Tem (based on analyses by Zakari Tchagbale).

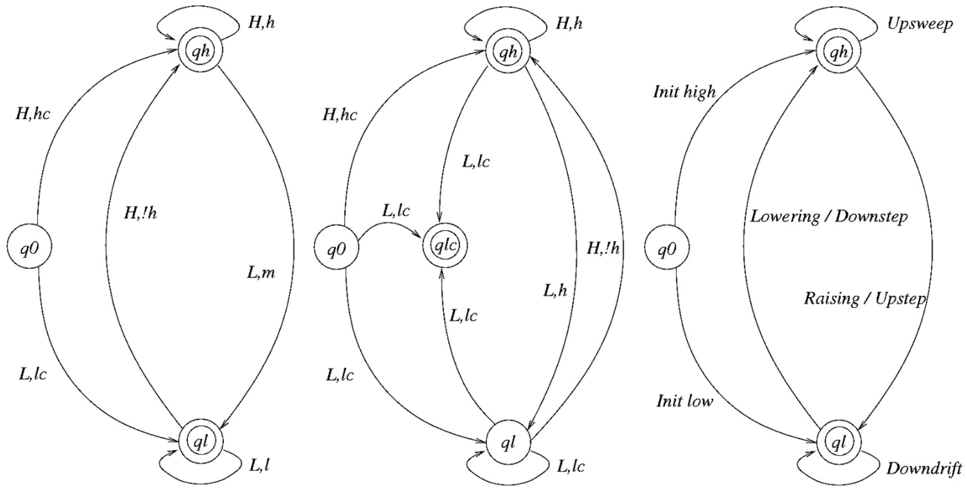


Figure 1: *Ewe, Tem and schematic FS networks.*

The Ewe FS network shows the basic form of a *start state* and an oscillation between two other states, one for each tone, in this case a *high state* and a *low state*. The automaton abstracts away from many aspects of the Ewe tone system, but it clearly represents the tone sandhi effects in Ewe H-L sequences, 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. The Tem FS network is a little more complex, in that a final low is explicitly included in the form of a "low constant" tone (it will be necessary to add this feature *addendum* to networks for many languages of course, perhaps all).

This automaton type can be generalised to other tone systems, as shown in the schematic FS network. Transitions between the three states, standing for "startup", "high" and "low", suffice for defining the main contexts for allotone mappings of the types attested in the literature; the terminology which is familiar from the literature is formulated as labels on the relevant transitions:

- *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 (upglide) high sequences are defined (I refer to this kind of pattern as *demiterrace drift*);
- *high-low*: transition from high to low, in which the first low of a sequence may be assimilated (raised) to mid or high.
- *low-low*: low *demiterrace*, in terms of which low constant or low downdrifting tones are defined (the other case of *demiterrace drift*);
- *low-high*: the context in which *automatic downstep* (and also assimilation of high to low) is defined, or, if floating low tones are included and may be taken to have the properties of overt tones, non-automatic downstep too;
- a 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 (I refer to terrace-level patterns as *terrace drift*).

The answer to the question of how to model a super-high tone at the end of a sequence of high tones in this type of model is left as a pleasant puzzle for the gentle reader.

The known facts for Baule are more complex. The relevant contexts are longer: it is not only adjacent contexts which are relevant for tone mapping constraints, but longer sequences. These constraints may be modelled by an additional oscillation in the model in order to accommodate the more complex contexts; this is modelled by the inner loop in the Baule network.

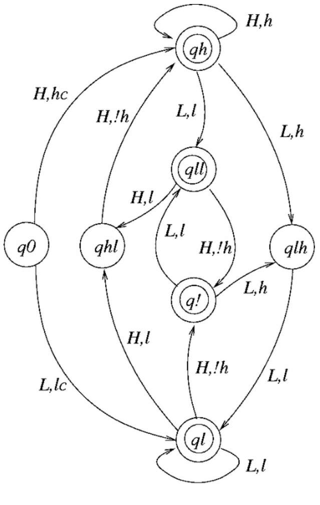
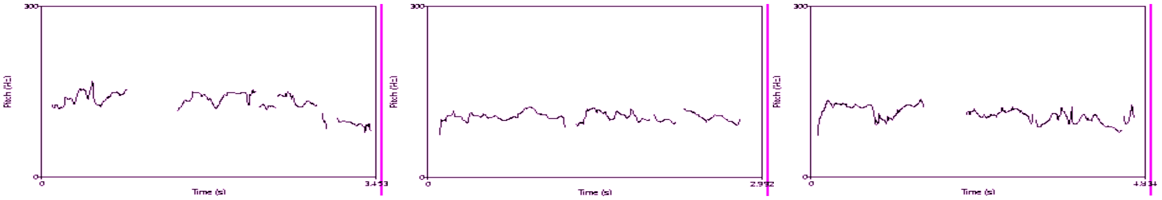


Figure 2: *Baule FS network.*

So far we have considered two-tone systems (the Ewe description abstracts away from mid tone and only models H and L sequences). 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 two 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, high and low demiterraces 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, which will need to be modelled as in the Tem FS network. Pitch traces of three consecutive Ega utterances are shown in the figure; the left hand side of the next figure shows the waveform and pitch trace of the first utterance with segmental and tonal annotation.

But we have already noted that Ega apparently has a discrete level tone system, therefore the motivation for an automatic downstep transition is absent. And so far, we have no evidence yet for other contextual effects (though this is not to deny 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 variation in the tone realisation mapping are found, then this extremely simple freewheeling model will need to be expanded again, with the introduction of further states in order to provide more transitions as contexts for the constraints.

Figure 3: *Three consecutive Ega utterances, illustrating discrete level tone realisation.*



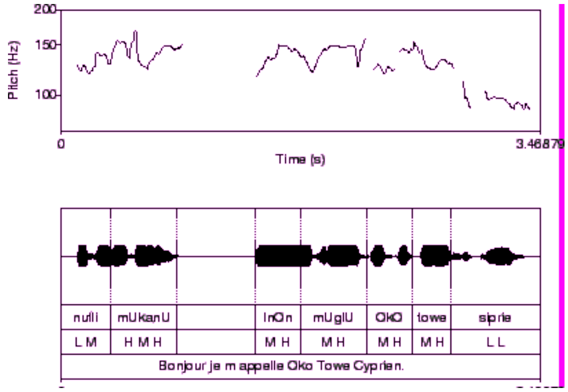


Figure 4: *Annotated Ega utterance.*

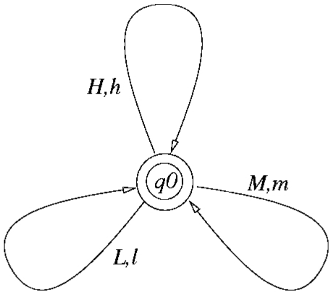


Figure 5: *Freewheeling 3-tone FST.*

6. Descriptive adequacy revisited: adding non-tonal trigger constraints

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. As noted at the outset, the following kinds of issue arise in descriptions of tone systems:

- *formalisms for tone*: representations of specific units, generalisations (rules), algorithms;
- *tone inventories*: 2-tone vs. >2-tone systems, features (African?) vs. units (Asian?), height vs. register; segmental vs. suprasegmental feature;
- *tone mapping* (realisation) operations: intrasegmental (assimilation, dissimilation, e.g. raising), intersegmental (spreading/dumping, displacement), downtrends (terraced vs. discrete level, terrace downdrift, demi-terrace downdrift, automatic and non-automatic downstep), uptrends (upglide, upsweep, upstep);
- *tone mapping domains*: phonological (neighbouring syllables, words), morphosyntactic (simplex and complex word, noun phrase, larger domains);
- *tone mapping triggers*: (near-)neighbouring tone, licensing vs. blocking consonant, morphosyntactic category.

The present contribution is concerned with the first three; current further work is addressing the last two.

The methodology for addressing the last two points is also adopted from finite state phonology and involves the working assumptions that

- phonological constraints can be formulated in terms of finite state transducers;
- this includes not only tonotactic (tone sequence) and the associated tone mapping constraints, but also the full range of tone trigger, from tone-tone constraints to segmental and morphosyntactic constraints;
- the finite state transducers can be *composed* into a single large (if unedifying) finite state transducer (Kay & Kaplan 1994), demonstrating the tractability of the approach and the overall FS properties of the prosodic system.

In other words, the claim is that contexts containing depressor consonants, or morphosyntactic domain information, can be separately formulated as finite state transducers, which can then be automatically combined by means of a calculus operating over finite state systems. This has been demonstrated for general Optimality Theory problems by Karttunen (1998), and the method appears to be directly applicable to prosody. This is the limit of comment on the problem in the present contribution, however.

7. Observational adequacy revisited: quantitative FS models

Quantifiable FS models of prosody have been well known for some three decades; the most well-known are the intonation models of Fujisaki, the Dutch IPO group, Pierrehumbert, with quantitative applications of the latter by Pierrehumbert & Liberman.

In recent work, Urua, Gut and I report on a production experiment and a new computational model in an attempt to resolve the question of the status of floating Low tones in Ibibio. Ibibio is a Lower Cross (Delta Cross) language of the (New) Benue-Congo family of languages and is spoken by over four million people in Akwa Ibom State and to a lesser extent in Cross River State of Nigeria. Ibibio is a classic example of a terraced level tone language (Welmers 1973) and has been shown to have two level tones, High and Low and a downstepped High tone. The downstepped High tone is usually preceded by a High tone, but can also occur initially. Other tonal realisations are the High-Low and Low-High contour tones, which are synchronically phonetic combinations of the level tones and may diachronically be from lowering and raising of the level tones, High and Low, respectively by the oral stop consonants, syllable loss, tonal assimilation, melody levels, etc.

In this study, we examine the phonetic properties of the following prosodic patterns: What are the baseline offset, pitch range and downdrift factors? What are the phonetic realisations (allotones) of High (H) and Low (L) tones in different tonal contexts? Is an overt Low tone in Ibibio similar to a floating Low tone? Do they cause the same degree of lowering? The computational model was designed in several stages, starting on the lines of the model introduced by Liberman and Pierrehumbert (1984).

We investigated several models, including overlay and sequential types, and came to the conclusion that overlay and sequential models are formally equivalent (cf. Gut & Gibbon, 2000). The final model contains a reference line of variable slope (exponential or asymptotic), and a multiplicative phonetic tone factor for each phonetic tone, as defined by an FS automaton for Ibibio which has the same topology as the FS automata for other two-tone languages.

The tone factors associated with the transitions in the FS network define the distance between the tones and the reference line, the factor being in general >1 for phonetic H tones and <1 for phonetic L tones (see below).

An exhaustive search algorithm is used, implemented in Scheme and running on a UNIX cluster. The search criterion, i.e. the best fit between the patterns calculated on the basis of the quantitative FS model and the data, is determined by the Average Magnitude Difference (AMD) function (the average of the differences between the absolute values of neighbouring pitches). The AMD was chosen mainly because the usual correlation functions are misleadingly high for the few data points concerned, but also because of its efficiency: it is additive rather than multiplicative, which makes the search procedure faster. This is important, since the complexity of the exhaustive search algorithm which was employed is exponential in the number of tone factors plus the reference line factor. In order to restrict the search space, initial investigations were made with a rather coarse granularity of model values.

The modelling technique reported in the previous literature (so far as descriptions of method are explicit enough to permit this kind of conclusion to be drawn) is to average paradigmatically over all the frequencies for one particular tone and then fit these syntagmatically to an overall pattern. This is not our approach. Rather, our technique is closer to the pattern recognition techniques used in speech technology, in that each individual utterance is optimally modelled syntagmatically, and paradigmatic generalisations are made subsequently. It is conceivable that the two approaches would yield the same or similar results, but this is by no means a necessary conclusion. Further details, and of course the descriptive results, are included in Gibbon, Urua & Gut (2000).

8. Computation: representations rather than procedures

So far I have not mentioned computers, processing, processes, procedures, algorithms, rules and the like. Neither have I mentioned representations explicitly. 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 of lexical prosody have had, in general, the following characteristics:

- concentration on representations of specific sequences, for example as autosegmental diagrammes, or as metrical trees;
- formulation of generalisations between specific representations as separate rules (e.g. downstep, assimilation);
- ignoring of the holistic properties of the representations into account;
- ignoring of the formal properties of the rules (are they processable, are they learnable);
- 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. So the point to be made here is that representations of *grammars*, not just of trees and lattices and matrices, need to be formulated, and that these grammars need to be computable by very general procedures. And it needs to be noted that rules, in the linguistic sense, are representations of linguistic generalisations, not procedures in the computational sense.

It is tempting to think that the contribution of computation lies in the use of computers. This is not true, though they are extremely helpful in enforcing rigour in developing models for theoretical approaches and in testing the models on large quantities of data. The main 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*:

- *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.
- *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:

- by derivation: a structure is derived by rules of inference from structural axioms;
- 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.

Nothing of what has been discussed so far pertains to the notion of algorithm, a term which tends to be 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. An algorithm has a certain well-defined complexity; of all the grammar types which one could postulate for tone grammars, algorithms for processing FS automata have the lowest complexity.

9. 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. How can tone systems be compared?

   b. 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?

   c. What are the general principles behind the different kinds of tonogenesis?

   d. How can the acquisition of tone systems be modelled - for instance with formal models for the automatic learning of FS automata?

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.

A promising direction is to examine the role which defaults play in finite state systems, and to combine the finite state modelling technique with a formalism in which to express preferences. This sounds like a tall order. 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 the underlying (lexical) patterns, coded in upper case characters, are the same. The outputs on the right hand side, coded in lower case characters, are different, corresponding to the different predictions made by descriptions of the different languages. The first equation shows an abstract, overly complex invented tone language, the second Baule, the third Tem. For the curious, the *default grammar,* coded in a preference logic formalism (DATR), is shown in the appendix.

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.

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 hope to have provided a useful intellectual tool taken from the Finite State Technology paradigm.

References

Gibbon, D. (1987). Finite State Processing of Tone systems. In: *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, Copenhagen. 291-297.

Gibbon, D. & Ahoua, F. (1991). DDATR: un logiciel de traitement d'héritage par défaut pour la modélisation lexicale. *Cahiers Ivoiriens de Recherche Linguistique* (CIRL) 27, 5-59.

Gibbon, D., Urua, E. & Gut, U. (2000). How low is floating low tone in Ibibio? Presented at the 30th Colloquium on African Languages and Linguistics, 27-30 August 2000, Rijksuniversiteit Leiden, NL.

Gut, U. & Gibbon, D. (2000). Is intonation linear? Proceedings of Prosody 2000, Cracow, Poland, pp. 83-88.

Carson-Berndsen, J. (1998). *Time Map Phonology: Finite State Models and Event Logics in Speech Recognition.* Dordrecht: Kluwer Academic Publishers.

Kaplan, R. M. & Kay, M. (1994). Regular models of phonological rule systems. *Computational*

Linguistics 20 : 331-378.

Karttunen, L. (1998). The Proper Treatment of Optimality in Computational Phonology. In: FSMNLP 98, International Workshop on Finite State Methods in Natural Language Processing, June 29 - July 1, 1998, Bilkent University, Ankara, Turkey

Schuh, R. G. (1978). Tone rules. In: Fromkin, V. (ed.) *Tone: A Linguistic Survey*. New York: Academic Press.

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 <>
        <>    == .
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```