Modelling, Formality and the Phonetics–Phonology Interface

Julian Bradfield

University of Edinburgh
jcb@inf.ed.ac.uk

ABSTRACT

Arguing for an increased cooperation between the fields of formal modelling and phonology, we illustrate the potential of several models from the computational sciences in phonology. We propose the skeleton of a multi-layer formal model from phonology through production and perception back to phonology.

Keywords: formal phonology, phonology-phonetics interface

1. FORMALITY

1.1. Why formalization?

The desirability of formalization in phonetics and phonology is not universally obvious to a general audience of phonetic scientists. On the other hand, those from the purer mathematical sciences, including mathematical linguistics, tend to take this desirability perhaps too much for granted. So we briefly discuss the advantages and pitfalls of formalization.

The most obvious advantage of a formal model is precision. A formal model (a structure described in mathematical terms, ultimately embeddable in a standard meta-mathematical universe such as set theory) leaves in principle no room for argument about what it does – although in practice a complex model may be too hard to analyse fully. The precision is useful in two ways: it forces clarity of thought, and it allows computational analysis and simulation. In linguistics formality has primarily been used for clarity, although there is an increasing trend towards computational simulations, which necessarily use formal models. In other fields relying on formal systems, there is an interesting difference between the natural sciences, where the prime validator of a model (e.g. the standard model of particle physics, the quantum mechanical model of atoms) is its ability to explain and yield predictions about reality; and the sciences of artificial systems, such as software design and verification, in which clarity of expression is the main validator. (The latter part of this claim may be debated; however, even in a field such as formal verification of software and hardware, where analysis and simulation is the motivating factor, it is widely quoted folklore that at least as many bugs are found due to the precision required by the act of formalizing, as are found by the subsequent analysis of the formal model.) Linguistics falls nicely between these two positions: on the one hand, language is a 'natural' system, about which we should be able to make predictions; on the other hand, language is an 'artificial' system arising from complex computational devices, and many arguments about competing theories are bedevilled by hidden assumptions and arguable interpretations.

However, there are dangers in the use of formal models. A pitfall common in the 'artificial' sciences is 'throwing the baby out with the bathwater': in the necessary process of abstracting to a level that one can deal with formally, one accidentally defines away the phenomenon one is trying to study or detect. The 'natural' sciences are less susceptible to such problems, as the validation against reality is usually more immediate, but it occurs even there. This pitfall is just one instance of the more general problem that the formal model has to be connected to reality, and this connection cannot itself be formal. In linguistics, the connection to reality is especially hard because even the surface realization of the real phenomenon is so complex.

1.2. In reply to 'Against Formal Phonology'

Port and Leary in a recent paper [10] mounted a focussed attack on the whole concept of formal phonology. In our view, this paper contains a number of misconceptions about the notion of 'formal system', resulting in false criticism. Reducing their argument to a somewhat caricatured nub, it is: "formal phonology is discrete, but speech is continuous – therefore formal phonology cannot accurately model speech production". Even more briefly: discrete and continuous time don't mix.

This is not a problem unique to linguistics – it has also occurred in the development of both the theory of computation and the practical modelling of systems with computational models. As Port and Leary repeatedly emphasize, the classical computer is in abstract a discrete digital device (though they curiously do not remark that every real computer is actually a continuous analogue device). Such was the model of computation from Turing onwards, while

continuous modelling was the realm of applied mathematicians wielding differential equations.

However, over the last couple of decades, an enormous amount of work has been done on combining the powerful toolboxes of classical automata and computation with the more 'realistic' real-time models, resulting in an entire field of 'hybrid systems'. This work is no less 'formal' than classical work; as well as producing beautiful theorems, it enables the formal modelling of many realistic real-time systems. Real time is not the only continuous metric brought into the computational fold; probabilistic and stochastic systems are equally studied.

Furthermore, there is another extension of classical computation with obvious application to phonological–phonetic processing. Namely, concurrency: the formal modelling of parallel or distributed processing. This is traditionally discrete, but can also be combined with real time. Concurrency underlies the ideas of autosegmental phonology, or even just feature bundles, and appears in simple form in the formal autosegmental model of Bird and Ellison [2]; it is also explicit in the overlappings of gestural phonology.

In the remainder of this paper, we argue that by drawing on the range of models developed elsewhere, we may progress towards many desiderata of phonological theory: a formal, discrete notion of phonology that yet acknowledges the individuality of cognitive categories; a process of transformation from phonology to real-time articulatory events; the re-analysis of perceptual events into cognitive categories; and the closing of the loop by changing phonological systems. Moreover, such models may be developed at many levels of abstraction, related by well understood (and formal) 'refinement relations', thereby possibly providing a plausible means of abstracting from individual, cognitive phonology to the traditional phonology of the last century.

2. SOME FORMAL MODELS WITH APPLICATIONS

In this section, we review briefly a selection of formal models, drawn chiefly from the computational sciences, and illustrate how they might be used in phonetics–phonology. We stress that the development of sound non-trivial formal models requires a great deal of time and specialist skills (modelling being more of an art than a science) as well as domain knowledge, and this article aims merely to demonstrate the potential of a wider range of such models.

At the basic level, all our students are familiar with the finite automaton and its equivalence with regular grammars. Most also know the Chomsky hierarchy of grammars, and their equivalence with increasing powerful classes of non-finite automata. Formal language theorists and computer scientists have produced a whole range of automata refining the Chomsky hierarchy. For example, Chomsky type 2 (context-free) grammars (generated by nondeterministic pushdown automata) may be restricted in several ways: the automaton may be determistic; the stack may be replaced by a simple counter. The whole theory of words generalizes to a (much richer) theory of trees, appropriate for considering the possible behaviours of non-determistic systems.

Most of the formal systems used in phonology, including rule-based systems and vanilla OT formalisms, fit within this classical part of formal computation theory.

2.1. Hybrid systems

The physical world, including the human speech apparatus, is largely continuous rather than discrete. The commonest model for continuous systems is systems of partial differential equations, which in all but simple cases are insoluble and so analysed by numerical methods. However, many human-constructed systems contain both discrete and continuous components, and with the increasing use of formal verification in recent decades, the study of hybrid formalisms, using both discrete and real-valued state variables, has become a large and popular field. One of the basic formalisms is the hybrid automaton of Alur et al. (see [6] for a review paper).

In brief, a hybrid automaton comprises a finite number of real-valued variables, a 'control graph' which is a finite automaton of 'control modes' (states) and 'control switches' (transitions). Each control mode has a 'flow condition', which is a (usually linear) first-order differential equation in the variables saying how they change during this control mode; and an 'invariant condition' giving conditions on the variables that must be true in this control mode (thereby forcing a control switch if the invariant fails). Each control switch has 'jump condition', giving a condition, in terms of the control variables and their rates of change, which enables (but does not force) the switch. There is an initialization condition giving the initial state and variable values.

Hybrid automata are attractive in control theory because not only are they powerful enough to express many control problems, they are well-behaved enough to be partially analysable by extensions of finite-automata-theoretic techniques – in particular, without solving all the differential equations.

In phonetics–phonology, hybrid automata are a natural candidate for the low-level formalization of Browman–Goldstein [3] gestural phonology, surely

one of the most intuitively attractive accounts of the articulatory side of phonology. The control variables are the positions of articulators, and the control modes correspond to the discrete phases of gestures (e.g. tongue tip moving to alveolum, resting there, moving away). Why 'low-level'? There is of course a lower level, concerned with neural activation signals of particular muscles, but there is also a more abstract level, on which we touch later, concerned with the causal and temporal arrangement of the gestural score.

2.2. Probabilistic and stochastic systems

Probability and statistics in phonology have a role that is hard to deny. Classical phonology is deterministic, and has never really coped well with the existence of marginal phonemes, sub-phonemic but still linguistically relevant distinctions, and of course the issue of phonologizing a gradient change. On the recognition side, categorizing noisy, gradient data is necessarily a probabilistic operation; on the production side, an attempt to make a sound is necessarily a probabilistic needle stuck in an achievable range – and if one adopts examplar-based theories of phonology, the target point itself may be probabilistically drawn from a space defined by the previously heard exemplars. Even with theories such as classical OT, which make strong claims to universality, some argue that one should include probabilistic, or at least non-deterministic, ranking of constraints (see Anttila and Cho [1], and also below).

Some probabilistic models are of course already extensively used in speech recognition, such as Hidden Markov Models, and these are tuned to learning of data. Several researchers, such as Coleman [4], Pierrehumbert (e.g. [4, 9]) and Goldsmith [5] have written on probabilistic phonology in production as well as perception; Pierrehumbert focussing on the examplar style, and Goldsmith developing a specific (simple, for demonstration purposes) theory to be compared with others such as OT. Here we are more concerned to mention the existence of formal models which can encode both classical and probabilistic phonologies, the latter interpreted broadly.

For many purposes, it suffices to take the natural extension of non-determistic automata to automata with probabilities attached to transitions, resulting in probabilities associated with finite behaviours, and a measure space over infinite behaviours. Although first studied in the 60s, many of the more sophisticated versions were developed more recently and are still the subject of extensive study – particularly the non-finite state automata, such as probabilistic pushdown automata and recursive Markov processes (related to the probabilistic context-free grammars used in computational grammar).

Further from the standard toolbox are probabilistic/stochastic *process algebras*. Process algebras, which in their basic form were invented about three decades ago, can be thought of as fundamental programming languages for multiple interacting computations. As such, they provide a structured and compositional way of building models of computational processes – for example, one might model the higher levels of production by processes for each articulator, giving commands to the lower hybrid automata level suggested above. The probabilistic versions of process algebras have been around for two decades, and would allow both the modelling of exemplar-based target selection, and higher level uses of probability such as selection of allophones, or selection of cases during phonemic split. (An introduction to one popular stochastic process algebra, which has found recent novel uses in biological and biochemical modelling, may be found in Hillston's [7].)

2.3. Concurrent formalisms

The earlier models of computation, including almost all those normally used in linguistics, are *sequential* – that is, computation or evolution of the system proceeds in a sequence of steps, whether it be the generation of a word by a finite automaton, the application of rules in a rewriting system, or the evaluation of an OT tableau. As we mentioned, in the presence of non-deterministic (or probabilistic) branching, one may consider instead the tree of possible behaviours; but each path through the tree is still sequential. Even the process algebras mentioned above (the classics being Milner's CCS and Hoare's CSP), which are explicitly designed to model distributed, interacting components, are inherently sequential: they *interleave* the actions of components.

Since the seminal work of Petri in the 60s, there have been models of computation which take seriously the idea of distributed computation, and in particular the idea that causal connection is not the same as temporal ordering, a distinction ignored by sequential models. Petri's model, now called *Petri nets*, is essentially automata with state that is distributed in space. There are also more recent models giving primacy to *events* and the causal relationships between them; and process algebraic languages for slightly higher-level 'programming'. These approaches are collectively known as 'true concurrency' (opposed to the 'fake concurrency' of interleaving) or 'partial order' (owing to the partially ordered causality relations) models.

There are several obvious parts of phonetics and phonology where true-concurrent models make evi-

dent sense. As we said in the introduction, autosegmental phonology is evidently an attempt to give a concurrent account of some phonological facts. The obvious example is Chinese (etc.) tones: tones go with syllables, not before or after the onset or rhyme. Ladd [8] expands on this to argue that partial precedence relations help to separate phonological and phonetic layers in the analysis of tones. More generally, each tier in an autosegmental representation runs independently until it synchronizes with another tier; at the most abstract level, ignoring real-time, this is a composition of communicating concurrent processes. It is possible to express autosegmental explanations of vowel harmony, assimilation etc. using true-concurrent models; however, our current version lacks elegance, and needs reconsideration – unless those who claim that autosegmentalism went too far with vowel harmony etc. are right!

Finally, let us remark that both at the production level (articulatory gestures) and the perception level (separate neural processing of different aspects of the signal), there is concurrent processing which then has to synchronize to produce the overall result.

3. RELATING SYSTEMS

A fundamental question in formal models is 'what does it mean for two models to be the same?'. In basic formal language theory, the usual answer is language equality, but in the richer models discussed above, there are many answers. There are more than a dozen possibilities even for plain sequential-but-nondetermistic computation, without considering real-time, probability or true concurrency. Likewise, the question of 'when does one system abstract another?' is also complex. A point in our programme where this issue becomes particular acute is the relation between the phonology of the individual and the phonology of the *langue*. While recognizing the fundamental importance of the cognitive phonology of the individual, we want to keep classical phonology, and perhaps the most obvious way to relate them is to ask for 'the phonology' of a language to be the finest abstraction of the individuals' phonologies. Abstraction is generally understood within any particular formal model, but the question of abstractions between different models requires, we understand, further development, before a question such as 'when does a concurrent rule-based phonology abstract from a set of exemplar-based individual phonologies?' can be answered with confidence. Indeed, even as we type this sentence, a colleague forwards a call for a conference on the same question in systems and software engineering!

4. CONCLUSION

We have outlined a variety of formal models, illustrating their potential in phonology, and we have claimed that such application will benefit phonology (and also formal computation, as some further work on their models and equivalences will be needed). We conclude by outlining the layered model we are developing (over the next several years).

phonology of the *langue* – concurrent model
individual phonology – probabilistic concurrent model
articulation (high-level) – concurrent real-time processes
articulation (low-level) – hybrid automata
sound – physics of vocal tract
sound – neural response
perception (low-level) – stochastic real-time automata
perception (high-level) – probabilistic concurrent model
individual phonology (via statistical learning)
phonology of the *langue* (via abstraction)

5. ACKNOWLEDGEMENTS

I thank the many people, both in Lingustics at Edinburgh and elsewhere, who have talked with me about these issues at various times. Special thanks go to Bob Ladd for particularly enlightening conversations.

6. REFERENCES

[1] Anttila, A., Cho, Y-m. Y. 1998. Variation and change in Optimality Theory. *Lingua* 104, 31–56.

[2] Bird, S., Ellison, T. M. 1994. One-level phonology: autosegmental representations and rules as finite automata. *Computational Linguistics* 20(1) 55–90.

[3] Browman, C.P., Goldstein, L. 1992. Articulatory phonology: an overview. *Phonetica* 49(3–4), 155–180.

[4] Coleman, J. S., Pierrehumbert, J. B. 1997. Stochastic phonological grammars and acceptability. *3rd Meeting ACL SIG in Comput. Phonology*, 49–56.

[5] Goldsmith, J. A. 2002. Probabilistic models of grammar: phonology as information minimization. *Phonological Studies* 5, 21–46.

[6] Henzinger, T.A. 2000. The theory of hybrid automata. In: Inan, M. K., Kurshan, R. P. (eds.) *Verification of Digital and Hybrid Systems*, 265–292.

[7] Hillston, J. E. 2005. Tuning systems: from composition to performance. *Computer J.* 48(4), 385–400, doi:10.1093/comjnl/bxh097.

[8] Ladd, D. R. 2007. Tone, autosegmental phonology, and the partial ordering of phonological elements. *CUNY Conf. on Precedence Relations*, January 2007.

[9] Pierrehumbert, J. B. 2003. Phonetic diversity, statistical learning, and acquisition of phonology. *Language and Speech* 46(2-3), 115–154.

[10] Port, R. F., Leary, A. P. 2005. Against formal phonology, *Language* 81(4), 927–964.