The complexity of phonology

András Kornai
MetaCarta Inc.
350 Massachusetts Avenue
Cambridge MA 02139
andras@kornai.com

1 Introduction

Two decades after the introduction of complexity-theoretic methods for the study of natural language grammars, there are still many technically incorrect arguments purporting to show the untractability of various formalisms. Here we analyze two such arguments, one old and one recent, both aimed at demonstrating that specific versions of phonological theory, as formulated in computational and theoretical work, are incapable of characterizing human phonology inasmuch as they lead to NP-hard[1] computational problems that are beyond the reach of any known fast algorithm. These demonstrations break down because they apply a mathematical technique, asymptotic analysis, whose major premises are not met. An alternative approach, Kolmogorov complexity, is proposed.

Barton (1986) studied the complexity of the analysis and generation problem of two-level phonology and morphology (TWOL) by means of encoding classical NP-hard problems (SAT and 3SAT) using the finite state transducer (FST) mechanism introduced independently in Koskenniemi's (1983) thesis and, with a somewhat different model, in Kaplan and Kay (1982, eventually published 1994). He concluded that TWOL systems are incapable of characterizing the human morphological ability because humans clearly perform the analysis and generation tasks in real time while the TWOL computational machinery must draw on exponentially growing resources as the problem size grows. Given that TWOL systems were at the time already the leading model of computational phonology and morphology, it is not surprising that the work drew the heated response *if it ain't broke, don't fix it* (Koskenniemi and Church 1988).

That such a response was not entirely unjustified is clear in hindsight. After two decades, TWOL remains the leading computational model of phonology and morphology (see e.g. Karlsson 2006), with working phonological/morphological grammars for hundreds of languages. If TWOL had the propensity to bump up against the complexity limit set by NP-hardness it would be nothing sort of miraculous that we can still use this framework without much trouble. Moore's law helps of course, but its main effect was to reduce the hardware required from the high-end workstations of the eighties to any cheap laptop today – the computations themselves didn't get notably more complex with increased lexical coverage and more detailed analyses.

Idsardi (2006a), building on earlier work (Ellison 1994, Eisner 1997, 2000) studied the complexity of Optimality Theory (OT) by means of encoding another classical NP-hard problem, directed Hamiltonian path search (DHPS), using the Gen/Eval mechanism of OT, and reached essentially similar conclusions about the suitability of OT for characterizing human phonology. Though Barton based his reduction proof on assimilation rules, while Idsardi uses dissimilation constraints to arrive at NP-hard problems, there is much that is common to these works, and here they are analyzed together, because they suffer from the same fundamental flaw: they use a mathematical method, asymptotic analysis, which relies in an essential fashion on the ability to create arbitrarily large problem instances. To some extent Barton, Berwick, and Ristad (1987:1.4.1) anticipated this criticism:

> Aren't then the complexity results irrelevant because they apply only to problems with arbitrarily long sentences and arbitrarily large dictionaries, while natural languages all deal with finite-sized problems?
>
> It is comforting to see that this argument explodes on complexity theoretic grounds just as it does in introductory linguistics classes. (...) Our goal is to determine the form and content of linguistic knowledge. (...) Once we have identified the principles that seem to govern the construction of sentences of reasonable length, there doesn't seem to be any natural bound on the operation of these principles. (...) If humans had more memory, greater lung capacity, and longer lifespan – so the standard response goes – then the apparent bound on the length of sentences would be removed.

As we shall see, domain length is indeed one of the parameters that delimit the complexity of combinatorial problems that can be encoded in phonology. But even if we grant arbitrary domain size, and even if we grant arbitrarily large dictionaries (a step that actually has considerable statistical support, see Kornai 2002), there are other limiting factors which ultimately block the appearance of large, computationally hard problems in phonology and morphology. Section 2 shows how these factors render the classical complexity-theoretic machinery powerless in the phonological domain, and Section 3 discusses why another set of mathematical tools, Kolmogorov complexity, is better suited to this domain.

2 Inventory Size

Idsardi (2006a) builds on earlier work (Ellison 1994, Eisner 1997), but stands out in this line of research by the exceptionally careful choice of the building blocks used in the proof. He demonstrates painstakingly that *self-conjoined constraints* forbidding the reuse of some element in a domain are widely attested in the phonologies of natural languages. Barton (1986) was less specific in this regard, but it is clear that the building blocks of his construction, *harmony rules*, are also widely attested – one would be hard put to find a phonologist interested in working with a lobotomized version of phonological theory that offered no means of describing such processes.

Both arguments can be summarized in the following three steps: (1) the computational/formal model incorporates mechanisms for handling assimilation (harmony) and dissimilation (self-conjoined) phenomena; (2) these mechanisms can be used to encode NP-hard problems and thus render the entire class of phonological problems NP-hard; and (3) NP-hardness is in evident contrast with human performance, which is linear, or even real time, as opposed to exponential, in the size of the input. Here we grant (1) and (2), and take issue with (3), as there is no evidence that machine performance on these tasks is exponential. There is no evidence because there *can be* no evidence, since the phonologies of natural languages are built on strictly finite, and as a matter of fact, remarkably small, inventories of combinatorial building blocks: features and phonemes.

Both for harmony rules and self-conjoined constraints the elementary building blocks are phonemes and the domains are words (strings built by concatenating phonemes). Idsardi (2006b) cites Swingley (2003) as an indication that there is more to phoneme inventory size p than meets the eye. Indeed, if we read Swingley's paper in an extreme way, we can conceive of a primal phonemic inventory wherein every potential feature contrast is kept, so that for some three dozen binary features we would obtain a set of 2^{36} or about 69 billion phonemes. This number is large enough to make us feel that asymptotic theory is of value even though strictly speaking the dataset is still finite. Now, given that all of these three dozen features are active in several languages, it is not at all clear why we do not have substantially larger phonemic inventories than we actually have, but, as Idsardi (2006b) notes, the cardinality p of attested phoneme systems tops out at around a hundred. Still, aren't a hundred distinct phonemes enough to make Idsardi's point that 100! permutations is not a tractable search space?

The answer depends on whether other limiting factors are present. There is a clear tendency of languages with larger p to have shorter words (Nettle 1995), and if domain size is limited to some small n, the maximal DHPS problem that could be encoded will be limited to this n even if p is large. For example, if the language permits only simple nuclei, trisyllabic or shorter stems, CC onsets but only C codas, the maximum stem length will be 12, and no amount of self-conjoined constraints over stems could encode a directed graph with more than 12 nodes. A search space of size 12! is no longer that formidable, and it is worth emphasizing that many of the classical morpheme structure constraints adduced by Idsardi operate over strictly bounded domains: for example, the Semitic point of articulation constraints typically over trilateral (rarely quadrilateral, and never, say, septilateral) roots, or Grassman's Law over two adjacent syllables.

So far, we have seen that problem size is bounded by $\min(n, p)$, where n is the length of the longest string potentially present in the domain and p is the size of the phoneme inventory. But there is a third, highly relevant limiting factor, 2^r, where r is the number of self-conjoined constraints in play. To see what is involved here, consider one of Idsardi's examples, Lyman's Law, which disallows the occurrence of more than one voiced obstruent per morpheme. We will allow morphemes (or prosodic words) of arbitrary size, and allow a fictional Japanese that has considerably more than 20 consonants and 5 vowels. What we do not allow, in this example, is more than two constraints, Lyman's Law and deaccentuation. This renders our generosity in regard to phoneme inventory or domain size irrelevant. The largest problem we can encode must be based on the distinctions voiced obstruent v. all other consonants and accented vowels v. unaccented, for a total of four graph nodes.[2]

Our final bound (still generous, as it assumes no further phonotactic restrictions that could mess up the encoding) is $\min(n, p, 2^r)$, and it is an interesting question why this number is so low in all phonologies. This puts constraints like *Repeat(stem) (proposed in Yip 1995 and discussed in Idsardi 2006b, where a number of other self-conjoined constraints are considered) in a different light. The actual resources to verify the satisfaction of a single constraint of this sort are *linear* in the size of the domain, and a word that contains all stems of the language (a full permutation by *Repeat) is inconceivable.

On the whole, phonology countenances very few unbounded domains. The typical situation is when larger structures are built out of the basic building blocks via intermediate structures. We do not pass from phonemes to words in a concatenative fashion – what we see is syllables (quite possibly with their own internal structure, but at any rate a very bounded and finitistic domain), feet, and perhaps cola, as intermediary structures. To the extent unbounded structures (e.g. unbounded feet) are sanctioned, the only dissimilatory effects that we see over these are restrictions pertaining to a finite (and very small) inventory of nodes in the feature geometry, which is by definition incapable of encoding arbitrarily many distinct units. But if we cannot have an arbitrary size inventory, we cannot encode an arbitrary-size directed graph, and the whole machinery of complexity theory loses its grip – all we have is a finite class of size-limited problems. The same reasoning applies to Barton's (1986) demonstration: assimilatory effects are observed on the same small inventory (nodes in the feature geometry) that is limited by substantive universals.[3]

Barton's reasoning was that TWOL systems are NP-hard to run while natural language words are easy to analyze, so TWOL systems do not do enough to sufficiently constrain the class of natural languages. The Koskenniemi-Church response essentially embraces this logic, saying in effect that TWOL systems lacking a proviso that there are not too many harmony processes operating in parallel are indeed hard to run, and therefore the required characterization of natural languages must use TWOL systems *with this proviso*. Here we embrace Idsardi's logic in the same fashion: OT without a proviso for small inventories and not too many dissimilation processes operating in parallel is NP-hard, and as such it must be an insufficient characterization of natural language phonologies.[4]

One can turn around the argument and ask *why* natural language phonologies show a strong limitation on $\min(n, p, 2^r)$? Perhaps a reason can be found in the fact that without such a limitation the system would manifest an exponential blowup that would threaten the efficiency of the mental computation. This is, of course, purely speculative. In reality we do not know why phoneme inventories are so drastically simplified compared to the proven capabilities of the perceptual apparatus, or why large numbers of unbounded assimilation or dissimilation processes do not operate in parallel. Yet the empirical fact remains that in the study of hundreds of phonologies not one computationally hard example has been found: the decision problem whether a particular string is licensed by a grammar describing actual phonological/morphological phenomena in natural language (as opposed to artificial problems created to trip up the mechanism) is quite easy. There remain many unsolved problems elsewhere in the system, in particular the mechanism required for the automatic acquisition of the computational building blocks from learning data is not yet fully understood, but the generation and recognition mechanisms are not under computational stress.

3 Kolmogorov Complexity

What the reduction proofs from Barton to Idsardi show are the limits of computational complexity theory rather than the limits of phonological theory: clearly, asymptotic analysis cannot be the mathematical tool of choice in a domain where arbitrarily large problem instances cannot be created at will. Since the substantive limitations on the number of units and constraints licensed by phonology render the standard asymptotic theory inapplicable, here we would like to call attention to another mathematical approach to the problem, Kolmogorov complexity, which does not lose traction when confronted with finite datasets. We are not the first to propose using Kolmogorov complexity (see in particular Clark 1994, Rissanen and Ristad 1994), and in fact the driving idea, that the best theory is the shortest theory, can be traced back to Pāṇini (see Kornai 2008: Ch. 7).

The *Kolmogorov complexity* of an object is defined as the length of the shortest program that generates it. Readers familiar with different programming languages (abstract models of computation) will immediately object that program length depends a great deal on the choice of model, and readers familiar with recursive function theory will likewise object that the general problem of finding the shortest program capable of a given task is unsolvable. Yet as Solomonoff (1960) demonstrated, these difficulties are not insurmountable, and complexity can still be defined, up to an additive constant, by program length. The technical apparatus of Kolmogorov complexity (for an encyclopedic overview, see Li and Vitanyi 1997) is harder to deploy than that of classical asymptotic theory, but this is a price very much worth paying for avoiding the key pitfall of the classical method, reliance on arbitrarily large hypothetical examples in a finitistic domain.

Relative to a given model of computation, the question of what is the complexity of a given finite grammar is equally meaningful irrespective of whether the grammar in question generates a finite or an infinite set. Thus, given a variety of phonologies/morphologies generating the same set, we can (and indeed, should) base the decision which one to prefer on the size of the system. This size is determined by two components: the size of the lexicon, i.e. the irreducible information including the phonological content of the roots and stems and the diacritics they carry, and the size of the pattern component (rules or constraints) that yield the surface forms from the lexicon. In most theories the two are kept separate, but in TWOL it is possible to combine them in a single automaton. As any practicing linguist knows, the list and the pattern components can be traded off against one another. By leaving more examples unanalyzed, and thereby increasing the number and size of the list items, one can do away with some putative rules or constraints, and likely also the diacritics that control them. Conversely, by finding the right pattern we may be able to shorten the lexicon considerably. We speak of a *linguistically significant* generalization only if by introducing it in the pattern component we save more in the list component – it is the hallmark of the pseudo-generalization that it affords little economy elsewhere in the system.

At first blush it may appear that we are interested in finding, among a class of models, those that minimize the sum of the size of the list and the pattern components. But upon some reflection it becomes clear that we need to go beyond this simple-minded approach, because the two components, while to some extent fungible, play a very different role in language acquisition. The list component must be memorized from A to Z, while the pattern component may refer to general patterns that are available to the learner as genetic dispositions. The innatist metaphor is convenient, but strictly speaking unnecessary to articulate this point of view: as long as patterns are reused across languages, their complexity, in computer science terms, will be *amortized* over these cases and contributes to the description complexity of a given language only a small fraction of the complexity they would have in the absence of Universal Grammar (UG).

As we are concerned with quantitative comparison of models, it may be helpful to put some numbers on the foregoing considerations. A conservative estimate of the body of irreducible information stored in the lexicon (irreducible in the sense that UG will have little to say about the idiosyncrasies of particular words or set phrases) would be the size of an abridged dictionary, 20,000–30,000 words, each requiring a few hundred unpredictable bits to encode their morphological, syntactic, and semantic aspects, for a total of about, say, a megabyte. Pāṇini's verbal and nominal lexicons, the Dhātupāṭha and the Gaṇapāṭha (which taken together outweigh the Ashṭādhyāyī about two to one) are together only a tenth of this, about 100 kbytes. In contemporary systems, such as the spellcheckers shipped with OpenOffice.org (OO.o), the average dictionary size is about 120 thousand stems, and the average stem takes up about 3.5 bytes compressed, for a total less than half a megabyte.

Does this cover all information that needs to be stored in the lexicon? Most variants of generative phonology/morphology remain silent both on the phonetic details of pronunciation (even when these are to some extent rule-governed, as emphasized by much contemporary work in laboratory phonology) and on the semantic content of morphological operations. For the phonetics, an upper bound can be derived from the fact that speech compression is already within a factor of five of the phonemic bitrate (see e.g. Hoshiya et al 2003), so the entire lexicon, including both phonetic and phonological information, is unlikely to exceed a couple of megabytes. While it is intuitively clear that a fair amount of information in the lexicon is semantical, in the absence of an agreed-upon model of lexical semantics it remains hard to quantify exactly how much. Assuming, for want of a better estimate, that about the same amount of semantic information is present in the lexicon as phonetic/phonological/morphological, we obtain an upper bound of 4 megabytes.

Having rather generously estimated the size of the lexicon at 4MB, of which no more than 10% are devoted to storing strictly phonological and morphological, as opposed to phonetic or semantic, information, let us now turn to the size of the grammar. Perhaps the largest, most systematic, and best compressed description of language-specific patterns ever devised, the Ashṭādhyāyī, weighs in at about 50 kilobytes. This is remarkably small, especially in light of the fact that it covers a great deal more than the phonology and morphology of the language and makes no use of a background theory of UG. It is a measure of the depth of the analysis Pāṇini provides that the size of his grammar is more than a third of the size of his lexicon. In the modern spellcheckers, among the hundred or so languages supported by OO.o there are only a handful where the (compressed) size of the pattern store is more than 10% of the (compressed) size of the lexicon, and even these are often indicative (e.g. in the case of the Coptic) of a less than fully developed lexicon.

Since the pattern component is so small compared to the list component, and since it amortizes well over different languages given a practical theory of UG, the primary focus of the effort in optimizing the system must fall on the list component. Here an important unresolved problem is whether the encoding should be tailored to minimize the simple or the frequency-weighted average of code length. The latter quantity, known as the *entropy* of the system, is what is taken as more relevant in contemporary computational work, while the tradition of generative phonology is more concerned with the former. In particular, once the observed frequencies of the surface forms are taken into account, recursive patterns like compounding no longer require a fully generative treatment in that systems with depth-limited recursion (see Mohri and Nederhof 2001), or even systems with no recursion, can approximate the frequency weighted set to an arbitrary degree of precision. The complexity of the pattern component, though numerically smaller, is made more important when comparing alternative models within the same theoretical framework in that the vast bulk of irreducible information in the lexicon is non-negotiable – each alternative has to account for it the same way, and it is only the differences across list sizes that matter.

How is, then, the complexity of the pattern component to be measured? Finite state transduction already offers a model that is sufficient to carry out phonological/morphological computations, and the TWOL technology is entirely neutral whether we conceive of these computations as SPE-style sequentially ordered rules (Kaplan and Kay 1994), as parallel rules (Koskenniemi 1983), or as OT-style ranked constraints (Frank and Satta 1998, Karttunen 1998). Extensions of the basic FST mechanism both to the multi-tiered and the frequency-weighted cases are well understood. Since (weighted, autosegmentalized) FSTs imply no commitment as to the nature of the specific phonological/morphological theory we adopt, we propose to take these as our general computational model: TWOL is suitable for this purpose precisely because it is devoid of UG content. The complexity of a model in this framework can be measured for example by the number of transducer states or arcs, or simply by the number of bits it takes to encode a transducer. One significant advantage of this method is that the overall transducer contains both the pattern and the list component, so theories that do not make a significant boundary between the two are accommodated with the same ease as theories that do.

OT, in contrast to TWOL, comes with a great deal of UG content. Ideally, it specifies in advance all constraint schemas, so that any possible grammar can be given by fixing the parameters of the constraints and specifying a ranking. For example, to fix the parameters of a (dis)harmony constraint will require only a few dozen bits that specify which features are involved and over what domain. To specify the ranking of N constraints requires $N \log(N)$ bits, so the overall compression effected by OT is quite significant: it is easy to define a system of a few dozen constraints and their ranking in one kilobyte that will take automata with millions of states and hundreds of thousands of transitions to implement. Generative grammar focuses on UG because only theories with a high degree of universal content can hope to account for the relative ease and speed of language acquisition. From the perspective of Kolmogorov complexity, UG emerges as the central device of data compression, enabling significant (perhaps as much as three orders of magnitude) data reduction.

4 Conclusions

Barton's prediction that TWOL must be overwhelmed by an exponentially growing need for computational time and space resources turned out to be entirely false, and Idsardi's prediction that OT must suffer the same fate will fail in a similar manner, and for the same reason: phonology simply does not have the diversity of atomic elements (neither items not arrangements) required to encode arbitrarily large problem instances.

This is not to say that OT, with an increasingly complex toolchest possibly including stratal organization, sympathy, candidate chains and other devices, is guaranteed to forever stay in the computationally easy realm, but Barton and Idsardi put the cart before the horse: normally we start looking for exponential growth when we encounter computational difficulties on real data, rather than creating difficulties by investigating artificial problems. There was no evidence in 1986 that TWOL was computationally hard, and there is no evidence in 2008 that OT is.

Since taking infinite limits is a *sine qua non* of classical complexity theory, we must conclude that this theory is not the appropriate tool for studying what is, in the final analysis, a very finitistic domain. Fortunately, a more appropriate mathematical tool, very much geared toward the complexity of finite objects, already exists in Kolmogorov complexity, and here we took the first steps in describing how this tool can be deployed in the study of the generative lexicon, which we estimated to have less than 4 megabytes of information content overall, of which no more than 400 kilobytes are devoted to purely morphological/phonological content.

Whitney (1879:xiii) chided Pāṇini for casting the facts of Sanskrit into the highly artful and difficult form of about four thousand algebraic-formula-like rules in the statement and arrangement of which "brevity alone is had in view." Today, with our focus on UG, we may not find this such a bad idea after all.

References

Barton, G. Edward. 1986. Computational complexity in two-level morphology. *Proc. ACL 1986* 53–59.

Barton, G. Edward, Robert C. Berwick, and Eric S. Ristad. 1987. *Computational complexity and natural language.* Cambridge MA: MIT Press.

Clark, Robin. 1994. Kolmogorov Complexity and the Information Content of Parameters. University of Pennsylvania: Institute for Research in Cognitive Science Technical Report 94-17.

Clements, George N. and S. Jay Keyser. 1983. *CV Phonology: A Generative Theory of the Syllable*. Cambridge MA: MIT Press.

Eisner, Jason. 1997. Efficient generation in Primitive Optimality Theory. In *Proc. ACL 1997* 313–320.

Eisner, Jason. 2000. Easy and hard constraint ranking in Optimality Theory: Algorithms and complexity. In *Proc. 5th ACL SIGPHON Meeting* 22–33.

Ellison, T. Mark. 1994. Phonological derivation in Optimality Theory. In *Proc. COLING 1994* 1007–1013.

Idsardi, William J. 2006a. A Simple Proof that Optimality Theory is Computationally

Intractable. *Linguistic Inquiry* 37:271–275.

Idsardi, William J. 2006b. Misplaced Optimism. ROA 840.

Frank, Robert and Giorgio Satta. 1998, Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24:307–315.

Hoshiya, Takahiro, Shinji Sako, Heiga Zen, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitantura. 2003. Improving the performance of HMM-based very low bit rate speech coding. *Proc. ICASSP 2003* 1:800–803.

Kaplan, Ron and Martin Kay. 1994. Regular Models of Phonological Rule Systems *Computational Linguistics* 20:331–378.

Karlsson, Fred. 2006. Kimmo Koskenniemi's first 60 years. In *Inquiries into Words, Constraints and Contexts. Festschrift for Kimmo Koskenniemi on his 60th Birthday*, ed. Anti Arppe et al. xiv–xvii Stanford CA: CSLI Publications.

Karttunen, Lauri. 1998. The proper treatment of Optimality Theory in computational phonology. In *Proceedings of the International Workshop on Finite-State Methods in Natural Language Processing* ed. Lauri Karttunen and Kemal Oflazer. 1–12. Ankara: Bilkent University.

Kornai, András. 2002. How many words are there? *Glottometrics* 4:61–86.

Kornai, András. 2006a. Is OT NP-hard? ROA 838.

Kornai, András. 2006b. Guarded Optimalism. ROA 841.

Kornai, András. 2008. *Mathematical Linguistics*. London: Springer-Verlag.

Koskenniemi, Kimmo. 1983. Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production. PhD thesis, University of Helsinki.

Koskenniemi, Kimmo and Kenneth W. Church. 1988. Complexity, Two-Level Morphology and Finnish. *Proc COLING 1988* 335–340.

Li, Ming and Paul Vitányi. 1997. *An Introduction to Kolmogorov Complexity and Its Applications*. London: Springer-Verlag.

Lieber, Rochelle. 1980. *On the Organization of the Lexicon*. PhD Thesis, MIT.

Mohri, Mehryar and Mark-Jan Nederhof. 2001. Regular Approximation of Context-Free Grammars through Transformation. In *Robustness in Language and Speech Technology* ed. Jean-Claude Junqua and Gertjan van Noord. 153–163. Dordrecht: Kluwer Academic Publishers.

Nettle, David. 1995. Segmental inventory size, word length, and communicative efficiency. *Linguistics* 33:359–367.

Rissanen, Jorma and Eric Ristad. 1994. Language acquisition in the MDL framework. In *Language Computations*, edited by Eric Ristad. 149–166. Americal Mathematical

Society DIMACS Series Vol 17.
Solomonoff, Ray J. 1960. A Preliminary Report on a General Theory of Inductive Inference. Report ZTB-138, Cambridge, MA: Zator Co.
Swingley, Daniel. 2003. Phonetic detail in the developing lexicon. *Language and Speech* 46:265–294.
Whitney, William Dwight. 1879. Sanskrit Grammar. Oxford: Oxford University Press.
Yip, Moira. 1995. Repetition and its avoidance: The case of Javanese. *Proceedings of South Western Optimality Theory Workshop*, edited by Keiichiro Suzuki and Dirk Elzinga, 238–263. Tucson, AZ

Notes

[0]This work supersedes the earlier critique (Kornai 2006a, 2006b) of Idsardi (2006a), and benefited enormously from Bill Idsardi's response at the time (Idsardi 2006b). The comments of the anonymous *LI* reviewers resulted in significant improvements in the final version.

[1]The notion of *NP-hardness* plays a key role in the mathematical theory of complexity: we say that a problems is NP-hard if it is at least as hard as any problem in the NP (Nondeterministic Polynomial) class. Hardness is measured by the growth in the number of elementary steps (Turing machine moves) needed for the computation as the size of problem instance increases. Technically, the NP class is defined by this growth being at most polynomial on nondeterministic Turing machines, but as a practical matter the celebrated P $\neq$ NP conjecture asserts that none of our algorithmic methods suitable for the relatively easy P (deterministic polynomial) class will generalize to the NP class. Since the NP-hard problems have withstood decades of efforts to find tractable (polynomial) algorithms for solving them, if a particular problem is NP-hard it is, at least to our current state of knowledge, intractable.

[2]An anonymous reviewer notes that the real Lyman's law in real Japanese requires special treatment of /p/, possibly opening the way for $r = 3$. This does not affect our main point, that encodable problem size is bound by the *minimum* of the limiting factors. Even $r = 4$ would imply a maximum of 16!, very far from the 100! that inventory size constraints alone would imply for a pseudo-Japanese with a larger phoneme inventory.

[3]An anonymous reviewer points out that diacritical (morpheme class) features can play the same role as phonological features. While this is certainly true, in general morpheme class features are fossilized remnants of earlier phonological features, and the diacritical use of phonological features is effected by placing standard features in nonstandard geometrical configurations (e.g. floating vowel features to trigger exceptional harmony processes, for h-aspiré see Clements and Keyser 1983, for conjugation classes of classical Latin see Lieber 1987). While this research program is not fully completed for all known cases of diacritic features, the implication is clear: the number of diacritics is limited to the number of pure phonological features times the number of nonstandard configurations, i.e. a small constant multiple of the former.

[4]While the overall line of argumentation is very much the same, the technical link between the Barton and the Idsardi arguments is more tenuous in that a full translation of OT to the finite state calculus assumes a bounded number of constraint violations, a criterion not necessarily met by the NP-hard constructions.