Guarded optimalism

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

In this note we offer a reply to Idsardi (2006b) responding to Kornai (2006), where we questioned the strength of the demonstration offered in Idsardi (2006a) that the decision problem whether a particular string is licensed by an OT grammar is NP-hard. Since there doesn't appear to be any disagreement between Idsardi's (2006a, 2006b) and Kornai (2006) as to the method to be followed in such a demonstration, we begin with disposing with the final rhetorical question *So what is the "it" that isn't broken?* – "it" is the central Gen/Eval mechanism of Optimality Theory, as currently understood by practicing phonologists everywhere. To be sure, this understanding is neither perfect nor entirely invariant under choice of phonologist, but there is enough coherence that we can dispense with the general philosophical arguments and move directly to the heart of the matter, especially as Idsardi was careful to build his demonstration on dissimilation phenomena that no phonologist would want to disregard.

There are two, strongly interconnected issues: the size of the domain on which the self-conjuction constraints operate and the size of the inventory involved in the phenomena. If either of these turns out to be finite, the standard definition of NP-hardness, where the complexity of a problem is defined as a function of the size of the input, loses its applicability. To make this point clear, consider chess, a game that is intuitively felt to be "hard". In the complexity-theoretic sense chess is *not* an NP-hard problem: there are only finitely many games (given the threefold repetition rule, itself a self-conjunction constraint of sorts), and in principle these could be enumerated, and the optimal strategy could be implemented by table lookup. For a problem to be NP-hard, one needs to be able to create arbitrarily large instances.

Idsardi (2006a) used phonemes as the inventory and words as the domain, and Idsardi (2006b) cites Swingley (2003) as an indication that there is more to phoneme inventory size 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 some value even though strictly speaking the dataset is still finite. Now, given that all of the three dozen features are active in some languages, it is not at all clear why we don't have substantially larger phonemic inventories than we actually have, but, as Idsardi notes, the cardinality p of attested phoneme systems tops out at around 100. 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 Hamilton path search that could be encoded will be limited to this n even if p is large. For example, if the language permits only trisyllabic or shorter stems, CC onsets but only C codas, the maximum stem length will be 12, and no amount of self-conjunction 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.

So far, we have seen that problem size is bounded by $\min(n, p)$, but there is a third, highly relevant limiting factor, 2^c, where c is the number of self-conjuction constraints in play. To see what is involved here, consider Lyman's Law, disallowing 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. Our final bound (still generous, as it assumes no further phonotactic restrictions that could mess up the encoding) is $\min(n, p, 2^c)$, and it is an interesting question why this number is so low in all phonologies. This puts constraints like *Repeat(stem) 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 inconcievable.

Here the situation is strictly analogous to Barton's (1986) demonstration that Kimmo systems can encode NP-hard problems, except Barton used assimilation rules (harmony processes) where Idsardi uses dissimilation rules. To revisit Barton's logic, his argument was that (i) Kimmo systems are hard to run and (ii) natural language words are easy to analyze, so (iii) Kimmo 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 Kimmo systems *without the proviso that there aren't to many harmony processes operating in parallel* are hard to run, and therefore the required characterization of natural languages uses Kimmo 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 hard, and as such it must be an insufficient characterization of natural language phonologies. Also, we must clearly admit we don't really know why phoneme inventories are so drastically simplified compared to the capabilities of the perceptual apparatus, or why large numbers of unbounded harmony or disharmony processes don't operate in parallel. We may speculate that these have to do with performance limitations, but in reality the relation between the number of logical operations (as counted by complexity theory) and the difficulty for a human to perform a deductive task (as measured by psycholinguistic experiment) is rather tenuous – people have a lot less difficulty with chess (requiring teraflop capacity in a machine) than with long division of two thousand-digit numbers (requiring a few dozen machine operations all told).

To summarize, we don't know what deeper reasons guard OT from blowing up computationally, but in practice it is evident that the decision problem whether a particular string is licensed by an OT grammar is not at all hard. What the reductions show are the limits of computational complexity theory rather than the limits of OT: there is reason for *guarded optimalism*.

References

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

William J. Idsardi (2006a) A Simple Proof that Optimality Theory is Computationally Intractable. *Linguistic Inquiry* **37** 271-275

William J. Idsardi (2006b) Misplaced Optimism. ROA

András Kornai (2006) Is OT NP-hard? ROA

David Nettle (1995) Segmental inventory size, word length, and communicative efficiency *Linguistics* **33** 359-367