A Simple Proof that Optimality Theory is Computationally Intractable

William Idsardi, University of Delaware

Abstract: Adapting arguments from Eisner (1997, 2000), this remark provides a simple proof of the NP-completeness of the generation problem for Optimality Theory (OT, Prince and Smolensky 2004) using only binary evaluation of constraints and constraints generally employed in the OT literature.

Key words: Optimaltiy Theory, computational complexity, NP-complete

1. Introduction

Eisner 1997 offers a proof that the generation problem for Primitive Optimality Theory is NP-complete. This proof involves constructing "an intermediate candidate set consisting of all permutations of *r* digits" (Eisner 2000: 29). Problems involving permutation sets are well known in the literature on computational complexity and intractability (see, for example Garey and Johnson 1979 and for specifically linguistic examples Barton, Berwick and Ristad 1987). However, Eisner's use of an idiosyncratic version of Optimality Theory and the unusual nature of some of the constraints employed in his proof have limited the impact of this important result. The present remark reconstructs Eisner's proof using only constraint-types attested in actual Optimality Theory analyses (e.g. Prince and Smolensky 2004, Kager 1999, Itô and Mester 2003). Furthermore, as suggested in Eisner 2000, we achieve this result assuming only binary evaluation of constraints (i.e. satisfied or violated once per candidate, no gradient evaluation of any sort).

2. Proof

The standard method of proof of NP-completeness is to encode a known intractable problem using the vocabulary and resources of the new problem. If there is a polynomial time reduction to the new problem, then the new problem is NP-complete, as explained in Garey and Johnson 1979, quoted in (1).

(1) "... once we have proved a single problem NP-complete, the procedure for proving additional problems NP-complete is greatly simplified. Given a problem $\Pi \in$ NP, all we need do is show that some already known NP-complete problem Π' can be transformed to Π." Garey and Johnson 1979:45.

The NP-complete problem to be transformed, as in Eisner 1997, is the Directed Hamiltonian Path problem (Garey and Johnson 1979: 60, 199, Cameron 1994: 167), (2). A Hamiltonian path is a tour of all vertices in a graph, without re-visiting any of the vertices.

(2) "A Hamiltonian path in a directed graph $G = (V, A)$ is an ordering of V as $<v_1, v_2, \ldots, v_n>$, where $n = |V|$, such that $(v_i, v_{i+1}) \in A$ for $1 \leq i < n$."
Garey and Johnson 1979: 60.

The trick now is to construct an OT grammar which computes Directed Hamiltonian Paths. We begin by taking an alphabet (i.e. a phoneme inventory) of n elements $I = \{$a, b, c, ... $\}$ representing the vertices of G in the obvious way (a = v_1, b = v_2, etc.). We then form an input word of n elements, /abc.../ (in fact any input word of n elements is sufficient). The infinite candidate set generated from this input by Gen is then evaluated by the following constraint hierarchy. I will present the constraints in a stratified hierarchy for ease of exposition; this is not in fact necessary for the proof as the ordering over the relevant constraints is immaterial.

In the first strata we have Max and Dep for segments, whose combined effect is to limit the candidate set to strings of the same length as the input string. This reduces the candidate set at this point to the finite set I^n which has n^n candidates. It is not necessary to have anything more than a binary evaluation of Max and Dep over the entire form to do this work. We do not even need to assume the unbounded evaluation of constraints in McCarthy 2003, quoted in (3).

(3) "... it is sufficient for any constraint to assign one violation-mark for *each instance* of the marked structure or unfaithful mapping in the candidate under evaluation."
McCarthy 2003: 130 (emphasis added)

Since there are candidates without *any* Max or Dep violations, it is unnecessary to assign more than one violation mark per candidate for the present purposes. It is not necessary to assign violation marks for each Max or Dep violation, as the candidate will be ruled out at the first such violation.

In the next strata we have the set of self-conjoined constraints (Itô and Mester 2003: 29) $*\alpha^2 = \{*a^2, *b^2, *c^2, ...\}$. Self-conjoined constraints forbid two instances of the same segment (or segment-type in the case of incomplete feature specifications) within the same domain, here the word (i.e. the whole candidate). Such constraints are found in Japanese (Lyman's Law) and Sanskrit (Grassmann's Law) and many other languages, as discussed by Itô and Mester (2003: 30, etc.). Again, these constraints can be evaluated in a binary fashion, as a single violation will be enough to rule out a candidate as there will be candidates without any violations of these constraints. At this point, the candidate set is now reduced to the set of strings of length n without repetitions, that is, the permutations of I, which has $n!$ candidates. We have now constructed a brute-force enumeration of all possible paths over V, regardless of whether or not the edges in the paths are elements of A.

In the next strata we have the constraints against including sequences of vertices not in G, that is, restricting the candidates to those with paths drawn from A. For this purpose, we have constraints of the form $*\alpha\beta$ where $\alpha = v_i$, $\beta = v_j$ and $v_iv_j \notin A$. Such constraints are the bread-and-butter of standard phonological accounts, as they bar certain sequences of segments, such as *NÇ̥ (Kager 1999, Itô and Mester 2003). All candidates remaining after this strata are Directed Hamiltonian Paths. Thus, we have transformed an NP-complete problem (finding Directed Hamiltonian Paths) into an Optimality Theory problem, and therefore have demonstrated that Optimality Theory is NP-complete. The transformation is obviously polynomial-time because the construction of the input is linear-time and the construction of the required constraint hierarchy is at most quadratic-time, the input having n elements, and the constraint hierarchy having at most $n^2 + n + 2$ constraints (n^2 possible sequence constraints, n self-conjoined constraints and Max and Dep).

3. Conclusions

In this remark, following the work of Eisner (1997, 2000), I have offered a simple proof that the generation problem for Optimality Theory is NP-complete. Under the current understanding of computational complexity results in which $P \neq NP$ (even though this hypothesis still evades proof), this makes Optimality Theory in general computationally intractable. In contrast, rule-based derivational systems are easily computable, belonging to the class of polynomial time algorithms, P (Eisner 2000: 32). The present proof used only binary evaluation of constraints and employed only simple OT constraints of types generally found in OT analyses of natural languages. The crux of the problem, as pointed out by Eisner 2000: 29 is the generation of an intermediate candidate set of permutations over an alphabet. We accomplished this with Max, Dep and the set of self-conjoined constraints $*\alpha^2$. Notice that it is not the initial infinitude of candidates that renders the problem computationally intractable, it is the exponential growth-rate in this relevant intermediate candidate space. Thus, any attempt to modify OT in order to render it computationally tractable will have to eliminate (or severely limit) the use of one of these constraint types, presumably the set of self-conjoined constraints. However, such restrictions must still somehow allow their use in the problems that Itô and Mester 2003 address, or invent new mechanisms to deal with effects like Lyman's Law and Grassmann's Law. But any such new mehanisms must be constructed so as not to be able to be used to construct permutation sets or to model other intractable computational problems.

References

Barton, G. Edward, Jr., Robert C. Berwick and Eric S. Ristad. 1987. *Computational Complexity and Natural Language.* Cambridge MA: MIT Press.

Cameron, Peter J. 1994. *Combinatorics: Topics, Techniques, Algorithms.* Cambridge: Cambridge University Press.

Eisner, Jason. 1997. Efficient Generation in Primitive Optimality Theory. *Proceedings of the 35th annual meeting of the Association for Computational Linguistics.* 313 – 320.

Eisner, Jason. 2000. Easy and Hard Constraint Ranking in Optimality Theory: Algorithms and Complexity. In J. Eisner, L. Karttunen and A. Theriault (eds.) *Finite-State Phonology: Proc. of the 5th Workshop of the ACL Special Interest Group in Computational Phonology* (SIGPHON), pp. 22-33.

Garey, Michael R. and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York: W. H. Freeman and Company.

Itô, Junko and Armin Mester. 2003. *Japanese Morphophonemics: Markedness and Word Structure.* Cambridge MA: MIT Press.

Kager, Rene. 1999. *Optimality Theory.* Cambridge: Cambridge UniversityPress.

McCarthy, John. 2003. OT Constraints are Categorical. *Phonology* 20: 75 – 138.

Prince, Alan S. and Paul Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar.* Malden, MA: Blackwell.