

# ON THE COMPLEXITY OF THE PROBLEM OF THE ACQUISITION OF PHONOTACTICS IN OPTIMALITY THEORY

GIORGIO MAGRI

**Abstract** — The problem of the acquisition of phonotactics in Optimality Theory is formulated as the problem of learning a ranking consistent with a finite set of data that furthermore corresponds to a smallest (w.r.t. set inclusion) language. The paper focuses on the *universal* formulation of the problem, whereby generating function and constraint set vary arbitrarily as inputs of the problem. It is shown that the universal problem of the acquisition of phonotactics in OT is not-solvable (NP-complete), even when we take the time to list all candidates. This result motivates a *non-universalistic* approach to the problem of the acquisition of phonotactics, whereby the problem is restricted to specific families of generating functions and constraint sets and solution algorithms are allowed to take advantage of the peculiar properties of these families.

## 1. INTRODUCTION

Research in language acquisition seeks to develop formal, explicit learning algorithms that provide good models of the actual learning strategies adopted by humans in acquiring their mother language. A naïve, straightforward approach to this goal consists of directly testing the modeling predictions of a variety of algorithmic schemes on actual acquisition data, in order to single out those schemes that provide the tightest model of the available data. Yet, the space of algorithmic possibilities is just too large to afford such a naïve approach. To overcome this difficulty, the following research strategy has proven useful. To start, the overall task of language acquisition is broken down into its basic components, stated as explicit formal problems. And the intrinsic complexity of these formal sub-problems is evaluated, with the tools of Complexity Theory. Under the plausible heuristic assumption that evolution has led to some form of computational optimality of the actual learning strategies adopted by human learners, algorithmic models are devised that provably solve the various formal sub-problems optimally, up to their complexity class. Finally, the predictions of various implementations of the optimal algorithmic schemes thus devised are tested on actual acquisition data, in order to fine tune the parameters and select the algorithmic implementations that offer a tighter model of the data. It is from this perspective that computational considerations gain currency within cognitive sciences. The resulting approach could well be called *Cognitive Computational Linguistics*. This very recent field differs from Computational Linguistics because of its *goal*, as it aims at cognitively plausible algorithmic models, rather than at efficient, approximate algorithmic solutions to practical problems. And it extends the boundaries of plain Linguistics because of its richer *methods*, as it construes the problem of language acquisition as a chapter of the theory of algorithms.

This paper contributes to the research agenda just sketched, by investigating the learning complexity of an important ingredient of the knowledge of the target language, namely knowledge of its *phonotactics*. The task of the acquisition of phonotactics is twofold. On the one hand, it requires learning which structures are *licit* w.r.t. the target phonotactics: for instance, English speakers know that [blik] would be a licit English word, despite the fact that it is accidentally unattested. On the other hand, it requires learning which structures are *illicit* w.r.t. the target phonotactics: for instance, English speakers know that \*[bnk] is unattested because it violates English onset cluster phonotactics. It is useful to investigate these two faces of the problem

---

*Date:* April 2011.

Earlier versions of this paper have been presented at NECPhon 2 (Yale University; November 15, 2008) and at SIGMORPHON 11 (University of Uppsala, Sweden; July 15, 2010). I wish to thank audiences at those venues for useful suggestions, as well as the SIGMORPHON's reviewers for detailed comments. I wish to thank in particular Adam Albright for lots of very useful discussion on the topic of this paper (and not only on that). This work was supported in part by a 'Euryi' grant from the European Science Foundation ("Presupposition: A Formal Pragmatic Approach" - P. Schlenker).

separately. I start with the former task, that is roughly formalized in terms of the *Consistency problem* (1). According to this problem, the learner has full knowledge of the underlying typology, is provided with a set of data sampled from a language in that typology, and is asked to find a phonotactics in the assigned typology that is consistent with those data. Note that the typology is allowed here to vary arbitrarily as an input to the problem. Following Heinz et al. (2009), I'll thus call (1) the *universal* formulation of the Consistency problem.

- (1) *Given:* a typology,  
           a finite set of data drawn from a language in the typology;  
*Find:* a grammar in the typology *consistent* with the data.

As just recalled, learning the phonotactics of a language involves both learning which structures are licit w.r.t. the target phonotactics as well as learning which ones aren't. The Consistency problem (1) only captures the former side of the learning task. Following a large literature, I formalize the latter side of the learning task by refining problem (1) with the additional *restrictiveness condition* (2b); see Berwick (1985), Manzini and Wexler (1987), Prince and Tesar (2004), and Hayes (2004), among others. According to this refined *Restrictiveness problem* (2), the learner needs to find a grammar which is not only compatible with the data by (2a), in order to rule-in licit structures (e.g. [blɪk] for English phonotactics); but is also restrictive by (2b), in order to rule-out illicit structures (e.g. [bnɪk]). Again, I focus on the universal formulation of the problem, whereby the underlying typology is allowed to vary arbitrarily as an input to the problem.

- (2) *Given:* a typology,  
           a finite set of data drawn from a language in the typology;  
*Find:* a grammar in the typology such that:  
     a) it is *consistent* with the data, as in problem (1),  
     b) its corresponding language is as *restrictive* as possible (w.r.t. set inclusion).

Section 2 provides a careful formulation of the Consistency problem (1) within the framework of Optimality Theory (henceforth: OT) and reviews the classical argument by Tesar and Smolensky (1998) that the Consistency problem in OT can be solved efficiently (i.e. in time that grows slowly with the complexity of the underlying OT typology) even in its universal formulation (i.e. without restrictions on the underlying OT typologies), provided we can afford the time to list all candidates. This discussion will set the stage for the analysis of the Restrictiveness problem (2). Section 3 provides a careful formulation of the latter problem within the framework of OT and states the main result of the paper, namely that the Restrictiveness problem cannot be solved efficiently in its universal formulation, contrary to the Consistency problem. This hardness result is due to the intrinsic complexity of the learning task, as it holds even if we give ourselves the time to list all candidates. In other words, the hardness result reported in this paper is orthogonal to other hardness results for OT available in the literature; see Eisner (1997), Idsardi (2006), Wareham (1998), and Heinz et al. (2009) for discussion. The proof of this hardness result is relegated to a final Appendix. Section 4 concludes the paper by discussing the implications of this hardness result for the field of Language Acquisition.

## 2. THE CONSISTENCY SIDE OF THE PROBLEM OF THE ACQUISITION OF PHONOTACTICS IS EASY

An OT typology is defined in terms of a 4-tuple of *typological specifications*, consisting of a set of *underlying forms*  $\mathcal{X}$ , a set of *surface forms*  $\mathcal{Y}$ , a *generating function* *Gen* mapping underlying forms into sets of surface forms, and a set of *constraints*  $\mathcal{C}$  assigning numbers of violations to pairs of underlying and winner forms. An example is provided in (3): the set of underlying and winner forms contains voiced and voiceless obstruents in onset and coda position; the generating function modifies obstruent voicing; and the constraint set contains faithfulness and markedness constraints related to obstruent voicing.

- (3) a.  $\mathcal{X} = \{ /ta/, /da/, /rat/, /rad/ \}$   
     b.  $\mathcal{Y} = \{ [ta], [da], [rat], [rad] \}$   
     c.  $Gen(/ta/) = Gen(/da/) = \{ [ta], [da] \}$   
          $Gen(/rat/) = Gen(/rad/) = \{ [rat], [rad] \}$

$$d. \mathcal{C} = \left\{ \begin{array}{l} F_{\text{pos}} = \text{IDENT}[\text{VOICE}]/\text{ONSET}, \\ F_{\text{gen}} = \text{IDENT}[\text{VOICE}], \\ M = *[\text{+VOICE}, \text{-SONORANT}] \end{array} \right\}$$

Let me denote an arbitrary *ranking* over the constraint set by  $\gg$ . A ranking of the constraints in (3d) is given in (4a): it sandwiches the markedness constraint in between the two faithfulness constraints, with the positional faithfulness constraint ranked at the top. I will also adopt the representation in (4b), where higher ranked constraints are placed at the top of the diagram.

$$(4) \quad \begin{array}{ll} \text{a. } F_{\text{pos}} \gg M \gg F_{\text{gen}} & \text{b. } \begin{array}{c} F_{\text{pos}} \\ | \\ M \\ | \\ F_{\text{gen}} \end{array} \end{array}$$

Let me denote by  $\text{OT}_{\gg}$  the *OT-grammar* corresponding to a ranking  $\gg$ , as defined in Prince and Smolensky (2004). The OT-grammar corresponding to the ranking in (4) is described in (5). Since  $F_{\text{pos}}$  is  $\gg$ -ranked above  $M$ , it lets /da/ surface faithfully. Since  $M$  is ranked above  $F_{\text{gen}}$ , it neutralizes the final voicing of /rad/.

$$(5) \quad \begin{array}{ll} \text{OT}_{\gg}(/ta/) = [\text{ta}] & \text{OT}_{\gg}(/da/) = [\text{da}] \\ \text{OT}_{\gg}(/rat/) = [\text{rat}] & \text{OT}_{\gg}(/rad/) = [\text{rat}] \end{array}$$

Let me denote by  $\mathcal{L}(\text{OT}_{\gg})$  the language corresponding to the ranking  $\gg$ , namely the set of those surface forms  $y \in \mathcal{Y}$  that are attainable through  $\gg$ , in the sense that there exists at least one underlying form  $x \in \mathcal{X}$  such that the OT-grammar  $\text{OT}_{\gg}$  maps the underlying form  $x$  into that surface form  $y$ .<sup>1</sup> The language corresponding to the ranking  $\gg$  in (4) is provided in (6).

$$(6) \quad \{[\text{da}], [\text{ta}], [\text{rat}]\}$$

Suppose that the learner is exposed to the language (6). Upon hearing many instances of the surface forms [da] and [rat], the learner will conclude that these forms do indeed belong to the target language. If the sets of underlying and surface forms coincide, then under mild conditions on the constraint set, it makes sense to assume that these surface forms are the faithful realizations of the corresponding identical underlying forms; see Tesar (2008) for discussion. Thus, the learner will safely assume that the ranking that corresponds to the target language validates the pairs of an underlying form and a corresponding winner surface form in (7a). Or perhaps the learner had access to some alternations and has thus noticed that the target grammar maps the underlying form /rad/ to the surface form [rat], thus neutralizing final devoicing. In this case, the learner might posit the set of underlying/winner form pairs in (7b).

$$(7) \quad \begin{array}{ll} \text{a. } D = \{(/da/, [\text{da}]), (/rat/, [\text{rat}])\} \\ \text{b. } D = \{(/da/, [\text{da}]), (/rad/, [\text{rat}])\} \end{array}$$

For what follows, it is not crucial how exactly the learner will get to a data set of underlying/winner form pairs such as those in (7), or which one of these two data sets the learner will actually posit. What is crucial is just that the learner has some device to construct a *data set* out of the target language, namely a finite set  $D$  of pairs  $(x, \hat{y})$  of an underlying form  $x \in \mathcal{X}$  and a corresponding intended winner candidate surface form  $\hat{y} \in \text{Gen}(x)$ . Throughout this paper, the only restriction on the data set  $D$  is that it is *consistent*, as stated in (8).

$$(8) \quad \begin{array}{ll} \text{a. A ranking } \gg \text{ is } \textit{OT-consistent} \text{ with a data set } D \text{ iff the corresponding OT-grammar } \\ \text{OT}_{\gg} \text{ corresponding to that ranking } \gg \text{ accounts for all pairs in } D, \text{ namely } \text{OT}_{\gg}(x) = \hat{y} \\ \text{for every pair } (x, \hat{y}) \in D. \\ \text{b. The data set } D \text{ is } \textit{OT-consistent} \text{ iff } D \text{ is consistent with a ranking according to (8a).} \end{array}$$

The core tenet of OT is that the set of all and only (possible) Natural Language phonologies coincides with the OT typology corresponding to some *actual* typological specifications  $(\mathcal{X}^{\text{act}}, \mathcal{Y}^{\text{act}}, \text{Gen}^{\text{act}}, \mathcal{C}^{\text{act}})$ . Suppose that these actual typological specifications were indeed known. The *Consistency problem* in OT could then be stated as in (9). The learner has available data

<sup>1</sup>In other words,  $\mathcal{L}(\text{OT}_{\gg})$  is the range of the function  $\text{OT}_{\gg}$ .

corresponding to some target language in the actual typology. These data consist of a set  $D$  of pairs of an underlying and a corresponding intended winner surface form. As we have no firm knowledge of which sets of such data the learner is able to extract, I assume here a formulation of the problem whereby the data set  $D$  is arbitrary. The only requirement on the data set  $D$  is that it be OT-consistent, in order for the problem to make sense. The learning task is to induce a grammar within the actual typology that is consistent with the data. As noted in Section 1, this Consistency problem formalizes one side of the task of the acquisition of phonotactics: learning of the phonotactically licit forms that they are indeed licit.

- (9) *Given:* a data set  $D$  of underlying/winner form pairs corresponding to some target grammar in the actual OT typology  $(\mathcal{X}^{\text{act}}, \mathcal{Y}^{\text{act}}, \text{Gen}^{\text{act}}, \mathcal{C}^{\text{act}})$ ;

*Find:* a ranking over the actual constraint set  $\mathcal{C}^{\text{act}}$  OT-consistent with the data set  $D$ .

We would like to list problem (9) among the interesting core sub-problems of the problem of the acquisition of phonology. But of course, problem (9) would only be interesting if we did have knowledge of the actual typological specifications  $(\mathcal{X}^{\text{act}}, \mathcal{Y}^{\text{act}}, \text{Gen}^{\text{act}}, \mathcal{C}^{\text{act}})$ . At this stage of the development of the field, of course we don't. Thus, problem (9) as it stands is of little use. A common strategy to overcome this difficulty is to switch from the initial statement of the problem in (9) to the variant in (10): instead of worrying what the actual typological specifications look like, we let the typological specifications vary arbitrarily as an input of the problem. This is of course not a small modification to the initial statement of the problem. This modification really makes sense only if we can confidently assume that a solution algorithm should not rely on properties of the specific, actual universal specifications. As we will see in the rest of this section, this assumption will turn out to be warranted and the switch from (9) to (10) harmless. This trick of coping with lack of knowledge of the actual typological specifications by quantifying universally over specifications, is standard in the OT complexity literature. For example, Eisner (2000) writes: “we follow Tesar and Smolensky (2000) in supposing that the learner already knows the correct set of constraints  $\mathcal{C}$ . The assumption follows from the OT philosophy that  $\mathcal{C}$  is universal across languages, and only the [ranking] of constraints differ. The algorithms for learning a ranking, however, are designed to be general for any  $\mathcal{C}$ , so they take  $\mathcal{C}$  as an input. That is, these methods are not tailored (as others might be) to exploit the structure of some specific, putatively universal [constraint set]  $\mathcal{C}$ .” Following Heinz et al. (2009), I call (10) the *universal* formulation of the OT Consistency problem.

- (10) *Given:* a) universal specifications  $\mathcal{X}$ ,  $\mathcal{Y}$ ,  $\text{Gen}$  and  $\mathcal{C}$ ,  
b) a finite OT-consistent data set  $D \subseteq \mathcal{X} \times \mathcal{Y}$ ;

*Find:* a ranking  $\gg$  of the constraint set  $\mathcal{C}$  that is OT-consistent with  $D$ .

The rest of this section summarizes Tesar and Smolensky's (1998) argument that problem (10) is “easy”, in the sense that there exists an algorithm that solves any instance of the problem *quickly*. Of course, how quickly an instance of the problem can be solved depends on the “size” of that instance: we should allow ourselves more time for instances of larger size, but be able to go fast on instances of smaller size; see subsection 5.1 for details. Thus, we need to complete the statement (10) of the problem with an explicit definition of the size of its instances. To this end, let the *cardinality* of the generating function  $\text{Gen}$  (on a data set  $D$ ) be the number  $|\text{Gen}(D)|$  defined in (11) as the cardinality of the largest candidate set over all underlying forms that appear in  $D$ .

$$(11) \quad |\text{Gen}(D)| = \max_{(x,y) \in D} |\text{Gen}(x)|$$

Following Tesar and Smolensky, I assume in (12) that the size of a given instance of problem (10) depends on three parameters: on the cardinality  $|\mathcal{C}|$  of the constraint set; on the cardinality  $|D|$  of the data set; and on the cardinality  $|\text{Gen}(D)|$  of the generating function (on the data set  $D$ ). It is uncontroversial that the size of a given instance of the problem should depend on the cardinalities  $|\mathcal{C}|$  and  $|D|$ . It is more delicate to let it depend on  $|\text{Gen}(D)|$  too. This means that a solution algorithm is allowed to take the time to list and inspect all candidates. The potential difficulty with this assumption is as follows: that  $|\text{Gen}(D)|$  could be very large, potentially exponential in the number of constraints  $|\mathcal{C}|$ ; thus, letting the size of an instance of the problem depend on  $|\text{Gen}(D)|$  might make the problem too easy, by loosening up too much the tight dependence on the number of constraints  $|\mathcal{C}|$ . This difficulty might indeed arise in the case of the original formulation (9) of the problem, where we do not have control over the generating function  $\text{Gen}$ . But this

difficulty disappears once that formulation (9) is replaced with the universal formulation (10): as we now require a solution algorithm to work for any constraint set and any generating function, it will also have to work for cases where the number of constraints  $|\mathcal{C}|$  is large but the cardinality of the generating function  $|Gen(D)|$  is small.<sup>2</sup> The one in (12) is thus the definitive formulation of the *(universal) OT Consistency problem*.

- (12) *Given:* a) universal specifications  $\mathcal{X}$ ,  $\mathcal{Y}$ ,  $Gen$  and  $\mathcal{C}$ ,  
 b) a finite OT-consistent data set  $D \subseteq \mathcal{X} \times \mathcal{Y}$ ;  
*Find:* a ranking  $\gg$  of the constraint set  $\mathcal{C}$  that is OT-consistent with  $D$ ;  
*Size:*  $\max\{|\mathcal{C}|, |D|, |Gen|\}$ .

Tesar (1995), Tesar and Smolensky (1998) and Tesar and Smolensky (2000, Ch. 7) (henceforth: Tesar and Smolensky)<sup>3</sup> prove the following important claim 1. This result had a profound impact on the field, for various reasons. *First*, because it represented the first explicit learnability result in OT. *Second*, because it provided a concrete case where no substantial harm comes from switching to the universal formulation of a learning problem, thus providing some justification for this technique. *Third*, because Tesar and Smolensky's proof rests on an elegant restatement of the Consistency problem (12) as a purely combinatorial problem, that has found widespread application in Computational OT. In the rest of this section, I summarize the two steps of their proof: the reduction of the Consistency problem to a combinatorial problem (Lemma 1 below); and the complexity analysis of the latter problem (Lemma 2 below).

**Claim 1.** *The (universal) OT Consistency problem (12) is tractable.*

After Tesar (1995) and Prince (2002), a *comparative tableau* is a matrix of the form (13), with  $n$  columns (one for every constraint) and an arbitrary number (say  $m$ ) of rows, whose elements are w's, l's and e's. I will say that the  $k$ th column of the tableau *corresponds* to the  $k$ th constraint  $C_k$ . I will denote an arbitrary comparative tableau by  $\mathbf{A}$ . I will also write  $\mathbf{A} \in \{L, E, W\}^{m \times n}$  to specify that it has  $m$  rows and  $n$  columns. I will call an arbitrary row of  $\mathbf{A}$  a *comparative row*;<sup>4</sup> I will often omit E's for the sake of readability.

$$(13) \quad \mathbf{A} = \underbrace{\left[ \begin{array}{cccccc} C_1 & \dots & C_k & \dots & C_n \\ W & L & W & L & E \\ L & W & W & E & E \\ E & W & W & L & L \end{array} \right]}_{n \text{ columns}} \left. \vphantom{\left[ \begin{array}{cccccc} C_1 & \dots & C_k & \dots & C_n \\ W & L & W & L & E \\ L & W & W & E & E \\ E & W & W & L & L \end{array} \right]} \right\} m \text{ rows}$$

Let me introduce next the new notion of *OT-consistency* in (14), as a combinatorial relation that holds between a comparative tableau and a ranking.

- (14) a. A ranking  $\gg$  is called *OT-consistent* with a comparative tableau  $\mathbf{A}$  iff, once the  $n$  columns of  $\mathbf{A}$  are reordered from left to right in decreasing order according to  $\gg$ , then the leftmost non-E entry of every row is a w.  
 b. A comparative tableau  $\mathbf{A}$  is called *OT-consistent* iff  $\mathbf{A}$  is OT-compatible with at least one ranking according to (14a).

With these preliminaries in place, we can now introduce the purely combinatorial problem (15). The input to the problem is a consistent comparative tableau; the task is to find a ranking consistent with that tableau; the size of an instance of the problem depends of course on the number  $n$  of columns and the number  $m$  of rows of the input comparative tableau.

- (15) *Given:* an OT-consistent comparative tableau  $\mathbf{A} \in \{L, E, W\}^{m \times n}$ ;  
*Find:* a ranking  $\gg$  that is OT-consistent with the comparative tableau  $\mathbf{A}$ ;  
*Size:*  $\max\{m, n\}$ .

<sup>2</sup>Furthermore, letting the size of an instance of the Consistency problem depend on  $|Gen(D)|$ , as well as on  $|\mathcal{C}|$  and  $|D|$ , immediately ensures that the (decision variant of the) problem is in  $\mathcal{NP}$ , namely that it admits a polynomial time verification algorithm.

<sup>3</sup>See also Eisner (2000) for some improvements on Tesar and Smolensky's argument.

<sup>4</sup>Prince (2002) calls a row of a comparative tableau an *elementary ranking condition* (ERC); Tesar and Smolensky call it a *mark data pair*.

There is a close relationship between the OT Consistency problem (12) and the combinatorial problem (15). To illustrate it, consider the instance of the OT Consistency problem corresponding, say, to the data set (7b). The first piece of data in this data set is the underlying/winner form pair (/rad/, [rat]). The *Gen* function in (3) provides only one corresponding loser candidate form, namely [rad]. We usually represent the relevant information in the form of the OT-tableau (16a). This representation encodes the actual number of constraint violations, as the number of stars in a given cell. Yet, because of the way the definition of an OT-grammar works, the actual number of constraint violations is not really needed. The information that we really need is just (16b): for every constraint, we just need to know whether it *prefers the winner* (namely it assigns more violations to the loser than to the winner) or it *prefers the loser* (namely it assigns more violations to the winner than to the loser) or it is *even* (namely it assigns the same number of violations to the winner and to the loser). Let me abbreviate (16b) as in (16c), marking each constraint with a W, an L or an E depending on whether it is winner- or loser-preferring or even.

$$(16) \quad \begin{array}{l} \text{a. } (/rad/, [rat], [rad]) \implies \begin{array}{|c|c|c|c|} \hline & /rad/ & & \\ \hline \text{a. } \llbracket & [rat] & & * \\ \hline \text{b. } & [rad] & & *! \\ \hline \end{array} \\ \\ \text{b. } \begin{array}{c} \text{winner} \\ | \\ (/rad/, [rat], [rad]) \\ | \\ \text{loser} \end{array} \implies \begin{array}{ccc} F_{\text{pos}} & F_{\text{gen}} & M \\ \text{“even”} & \text{“prefers the loser”} & \text{“prefers the winner”} \end{array} \\ \\ \text{c. } (/rad/, [rat], [rad]) \implies \begin{array}{ccc} F_{\text{pos}} & F_{\text{gen}} & M \\ \text{E} & \text{L} & \text{W} \end{array} \end{array}$$

If we adopt the same representation also for the other pair (/da/, [da]) in the data set  $D$  in (7b), then we end up representing the set  $D$  with the comparative tableau in (17). Its elements are all W's, L's and E's; it has as many columns as there are constraints; it has as many rows as there are relevant triplets of an underlying form, the intended winner and a corresponding loser. There is only one ranking OT-consistent with the data set  $D$  in (7b), namely the ranking  $F_{\text{pos}} \gg M \gg F_{\text{gen}}$  in (4). If the columns of the tableau (17) are ordered according to this ranking from left to right in decreasing order, then the leftmost entry different from E is a W in both rows, as required by (14) in order for the tableau to be consistent with this ranking. No other ordering of the columns of the tableau (17) has this property. In other words, we can find all and only the rankings that solve the instance of the Consistency problem corresponding to the data set  $D$  by solving the instance of the combinatorial problem (15) corresponding to the corresponding comparative tableau.

$$(17) \quad D = \left\{ \begin{array}{l} (/da/, /da/) \\ (/rat/, /rat/) \end{array} \right\} \implies \begin{array}{c} \text{winners} \\ | \\ (/da/, [da], [ta]) \\ | \\ (/rad/, [rat], [rad]) \\ | \\ \text{losers} \end{array} \implies \begin{array}{|c|c|c|} \hline F_{\text{pos}} & F_{\text{gen}} & M \\ \hline \text{W} & \text{W} & \text{L} \\ \hline \text{E} & \text{L} & \text{W} \\ \hline \end{array}$$

These simple considerations can be straightforwardly generalized. The data set  $D$  given with an instance (12) of the Consistency problem can be paired up with its *corresponding comparative tableau*  $\mathbf{A}_D$  defined as in (18), that generalizes the procedure illustrated in (17).

- (18) For every underlying/winner form pair  $(x, \hat{y})$  in the data set  $D$  and for every loser candidate  $y \in \text{Gen}(x)$ , construct a comparative row as follows:

$$\begin{array}{c} \text{winner} \\ | \\ (x, \hat{y}, y) \\ | \\ \text{loser} \end{array} \implies [ a_1 \quad \dots \quad a_k \quad \dots \quad a_n ]$$

$$a_k \doteq \begin{cases} \text{W} & \text{if } C_k(x, \hat{y}) < C_k(x, y) \quad (C_k \text{ prefers the winner } \hat{y}) \\ \text{L} & \text{if } C_k(x, \hat{y}) > C_k(x, y) \quad (C_k \text{ prefers the loser } y) \\ \text{E} & \text{if } C_k(x, \hat{y}) = C_k(x, y) \quad (C_k \text{ is even}) \end{cases}$$

Organize all these comparative rows one underneath the other in a comparative tableau with  $n$  columns and many rows (the order of the rows does not matter).

The number  $n$  of columns of the comparative tableau  $\mathbf{A}_D$  constructed in (18) coincides with the number  $|\mathcal{C}|$  of constraints. And the number  $m$  of its rows can be bound as in (19b). In fact, we get a comparative row for each one of the  $|D|$  pairs in the data set and for each one of the corresponding candidates, where the number of candidates is upper bound by the cardinality  $|\text{Gen}(D)|$  of the largest candidate set. Thus, an instance of the OT Consistency problem (12) can be transformed through (18) into a corresponding instance of the combinatorial problem (15) with comparable size. Note that this crucially hinges on the choice of letting the size of an instance of the OT Consistency problem generously depend not only on  $|\mathcal{C}|$  and  $|D|$  but also on  $|\text{Gen}(D)|$ .

$$(19) \quad \begin{array}{l} \text{a. } n = |\mathcal{C}| \\ \text{b. } m \leq |D||\text{Gen}| \end{array}$$

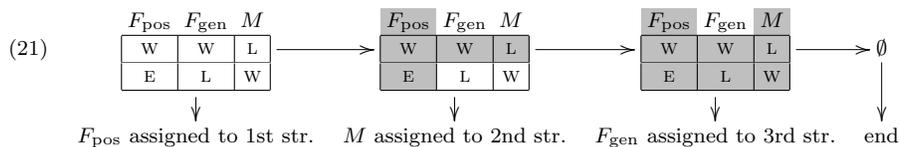
A straightforward generalization of the reasoning above (17) shows that a ranking  $\gg$  is OT-consistent with a data set  $D$  according to (8a) iff that ranking  $\gg$  is OT-consistent with the corresponding comparative tableau  $\mathbf{A}_D$  according to (14a). And a data set  $D$  is OT-consistent iff the corresponding comparative tableau  $\mathbf{A}_D$  is OT-consistent. In other words, (14) is a graphic description of the original notion of OT-consistency (8). And problem (15) is a combinatorial reformulation of the OT Consistency problem (12), in the sense that any algorithm that efficiently solves the former can be turned into an algorithm that efficiently solves the latter, as stated in the following Lemma. This Lemma represents the first step of Tesar and Smolensky's proof of claim 1: it ensures that, in order to prove that the original OT Consistency problem (12) is tractable, it is sufficient to prove that the combinatorial reformulation (15) in terms of comparative tableaux is tractable.

**Lemma 1.** *Consider an algorithm  $\text{Alg}$  that solves the combinatorial problem (15) efficiently. Consider then the algorithm  $\text{Alg}'$  that takes a data set  $D$ , constructs the corresponding comparative tableau  $\mathbf{A}_D$  and runs  $\text{Alg}$  on it, as stated in (20).*

$$(20) \quad \text{Alg}'(D) = \text{Alg}(\mathbf{A}_D)$$

Then the algorithm  $\text{Alg}'$  efficiently solves the OT Consistency problem (12).

Tesar and Smolensky next develop a simple solution algorithm to the combinatorial problem (15). Let me illustrate the idea with an example, as in (21). Suppose that the input tableau is the one constructed in (17). Our goal is to come up with an OT-consistent ranking  $\gg$ , according to (14a). This means that the tableau obtained by reordering the columns from left to right in decreasing order according to  $\gg$  has the property that the leftmost non-E entry of each row is a w. The top ranked constraint must head a column that does not contain a single L. In our case, the only such constraint is  $F_{\text{pos}}$ , that thus gets assigned to the top stratum. The constraint that can be assigned to the next stratum must head a column whose only L's belong to rows where the top ranked constraint  $F_{\text{pos}}$  has a w. In other words, it must head a column that does not contain a single L once we strike out the rows where the top ranked constraint  $F_{\text{pos}}$  has a w. In our case, the only such constraint is  $M$ , that thus gets assigned to the second stratum. The constraint that can be assigned to the next stratum must head a column that does not contain a single L once we strike out rows where at least one of the two top ranked constraints  $F_{\text{pos}}$  and  $M$  have a w. In our case, the only such constraint is  $F_{\text{gen}}$ , that thus gets assigned to the bottom stratum. As all constraints have been ranked, the algorithm stops.



The procedure just illustrated can be straightforwardly extended to a general comparative tableau  $\mathbf{A}$  with  $n$  columns as in (22). Step (22a) corresponds to the vertical arrows in (21); step (22b) corresponds to the horizontal arrows in (21). Algorithm (22) is called by Tesar and Smolensky *Recursive Constraint Demotion* (henceforth: RCD).

$$(22) \quad \text{Repeat } n \text{ times:}$$

- a. assign to the highest available place in the ranking a yet un-struck constraint whose column in  $\mathbf{A}$  does not contain any un-struck L;
- b. strike out every row of  $\mathbf{A}$  that has a w under the constraint just picked in step (a) and then strike out the entire column corresponding to that constraint.

If the input comparative tableau  $\mathbf{A}$  is OT-consistent, then RCD returns a ranking OT-consistent with  $\mathbf{A}$  in  $n$  steps. Furthermore, all rankings OT-consistent with  $\mathbf{A}$  belong to the search space of RCD.<sup>5</sup> And if the input tableau is not OT-consistent, RCD detects that, in the sense that it gets stuck before all constraints are ranked. Finally, RCD is efficient, as it repeats  $n$  iterations each of which takes at most  $nm$  time (the algorithm might need to scan  $n$  columns with  $m$  entries each). We have thus proved the following Lemma.

**Lemma 2.** *RCD (22) is an efficient solution algorithm for the combinatorial problem (15), which is therefore tractable.*

Lemma 1 guarantees that tractability of the combinatorial problem (15) ensures tractability of the Consistency problem (12). And Lemma 2 then ensures that the former combinatorial problem is indeed tractable. Claim 1 concerning tractability of the Consistency problem (12) thus follows from these two lemmas.

### 3. THE RESTRICTIVENESS SIDE OF THE PROBLEM OF THE ACQUISITION OF PHONOTACTICS IS HARD

As noted in section 1, the task of learning the phonotactics of the target adult language is twofold: the child needs to learn a grammar which is at the same time rich enough to be *consistent* with all phonotactically licit structures and furthermore *restrictive* enough to rule out all illicit ones. To illustrate, suppose again that the underlying OT typology is defined through (3) and that the target language the learner is exposed to is (6). Then, the acquisition of phonotactics can be broken down into the two tasks in (23).

- (23) Learn a ranking such that:
- a. is consistent with [ta], [da], [rat] (*Consistency task*);
  - b. is not consistent with [rad] (*Restrictiveness task*).

The first one (23a) of these two tasks is correctly modeled by the Consistency problem (12): no matter which data set (7) we consider, a solution of the corresponding OT Consistency problem will be consistent with the licit forms in the language. But what about the restrictiveness task (23b)? Suppose first that the data set is (7b), repeated in (24b). This data set contains in particular the underlying/winner form pair (/rad/, [rat]). Any ranking consistent with this pair will neutralize final voicing and will thus generate a language that does not contain [rad]. In other words, in this lucky case, the restrictiveness task (23b) reduces to the consistency task (23a), and the Consistency problem thus provides an adequate formalization of the entire task of the acquisition of phonotactics.

- (24) a.  $D = \{(/da/, [da]), (/rat/, [rat])\}$   
 b.  $D = \{(/da/, [da]), (/rad/, [rat])\}$

<sup>5</sup> The definition of RCD given in (22) is slightly different from the original definition by Tesar and Smolensky. The difference between Tesar and Smolensky's original definition of RCD and the one given here shows up for comparative tableaux that have more than one constraint with no undeleted L's. According to the definition (22), RCD arbitrarily chooses *one* such constraint and assigns it to the highest available stratum. According to Tesar and Smolensky's original definition, RCD assigns *all* such constraints to the highest available stratum and then outputs a total ranking which is an arbitrary refinement of the non-total ranking thus constructed. To illustrate how the two definitions differ, consider the comparative tableau in (i).

$$(i) \quad \begin{bmatrix} C_1 & C_2 & C_3 \\ w & e & l \\ e & w & w \end{bmatrix}$$

Tesar and Smolensky's original RCD first computes the non-total ranking  $\{C_1, C_2\} \gg \{C_3\}$  and then outputs one of its two refinements, namely either  $C_1 \gg C_2 \gg C_3$  or  $C_2 \gg C_1 \gg C_3$ . Thus, the ranking  $C_1 \gg C_3 \gg C_2$  lies outside of the search space of Tesar and Smolensky's RCD, despite the fact that this ranking too is OT-consistent with the given comparative tableau (i). Instead, the version of RCD defined in (22) might output such a ranking, provided that the algorithm chooses  $C_1$  at the first step,  $C_3$  at the next step and  $C_2$  at the last step. More generally, the version of RCD defined in (22) poses no artificial restrictions on the search space, which indeed contains any ranking OT-consistent with the given comparative tableau.

But things are very different in the case of the data set in (7a) repeated in (24a), consisting of faithful underlying forms only. The corresponding Consistency problem (10) admits two solutions, stated in (25) together with the corresponding languages. Only the solution (25b) solves the restrictiveness task of recognizing that [rad] is not phonotactically well formed.

$$(25) \quad \begin{array}{l} \text{a. } F_{\text{pos}} \gg F_{\text{gen}} \gg M \implies \mathcal{R}(\text{OT}_{\gg}) = \{\text{ta, da, rat, rad}\} \\ \text{b. } F_{\text{pos}} \gg M \gg F_{\text{gen}} \implies \mathcal{R}(\text{OT}_{\gg}) = \{\text{ta, da, rat}\} \end{array}$$

One option to cope with this difficulty would be to introduce assumptions to the effect of ruling out ambiguous data sets such as (24b), allowing only informative data sets such as (24a). This move would correspond to the assumption that the learner has indeed available a procedure that allows him to extract from the target language informative data sets such as (24a). But this assumption sounds implausible. As reviewed in Hayes (2004), phonotactics is acquired at an early stage of language development, when knowledge of morphology is plausibly still lagging behind. In the absence of morphological decomposition, the child lacks any evidence for phonological alternations. The safest learning strategy thus seems to posit faithful data sets such as (24b), that are provably consistent under mild assumptions on the the constraint set, as shown in Tesar (2008). Following a large literature, I will thus adopt an alternative strategy to cope with the difficulty just highlighted. The starting point is the straightforward observation that the target language (25b) is a proper subset of the incorrect language (25a). I thus switch to the more demanding variant of the Consistency problem (10) stated in (26). Again, the learner is provided with data drawn from some language in the actual OT typology  $(\mathcal{X}^{\text{act}}, \mathcal{Y}^{\text{act}}, \text{Gen}^{\text{act}}, \mathcal{C}^{\text{act}})$ , in the form of a set  $D$  of underlying/winner form pairs. As in the case of the Consistency problem, no restrictions are placed on the data set  $D$ , that could very well consist just of faithful pairs of underlying and winner forms. What is new is that the learner's task is not just to come up with *any* ranking OT-consistent with  $D$ , by (26a). Rather, (26b) now requires the learner to come up with one such ranking that furthermore has the property that the corresponding language is as small as possible (w.r.t. set inclusion), among the languages generated by rankings OT-consistent with the data  $D$ ; see for instance Angluin (1980), Berwick (1985), Manzini and Wexler (1987), Prince and Tesar (2004) and Hayes (2004), as well as Heinz and Riggle (to appear) for a review. I will call (26) the *OT Restrictiveness problem*.

$$(26) \quad \begin{array}{l} \textit{Given:} \text{ a data set } D \text{ of underlying/winner form pairs corresponding to some target gram-} \\ \text{mar in the actual OT typology } (\mathcal{X}^{\text{act}}, \mathcal{Y}^{\text{act}}, \text{Gen}^{\text{act}}, \mathcal{C}^{\text{act}}); \\ \textit{Find:} \text{ a ranking } \gg \text{ over the actual constraint set } \mathcal{C}^{\text{act}} \text{ such that:} \\ \quad \text{a) } \gg \text{ is OT-consistent with the data set } D; \\ \quad \text{b) there is no ranking } \gg' \text{ that satisfies (a) and such that the language corre-} \\ \quad \text{sponding to } \gg' \text{ is a proper subset of the language corresponding to } \gg, \text{ i.e.} \\ \quad \mathcal{L}(\text{OT}_{\gg'}) \subsetneq \mathcal{L}(\text{OT}_{\gg}). \end{array}$$

As already noted above in the case of problem (9), also the Restrictiveness problem (26) is of little interest, as long as we have no firm knowledge of the actual universal specifications. To overcome this difficulty, we switch from the original formulation (26) of the problem to the universal reformulation (27), that lets the typological specifications vary arbitrarily as an input to the problem. Here, I have complemented the statement of the problem with the specification of the size of its instances, which governs the time that a solution algorithm is allowed to take in order to solve that instance of the problem. This problem involves the language  $\mathcal{L}(\gg)$  generated by a ranking  $\gg$ , which in turn depends on the total number of forms  $|\mathcal{X}|$  and on the cardinality of the candidate set  $\text{Gen}(x)$  for all underlying forms  $x \in \mathcal{X}$ . Thus, it make sense to let the size of an instance of the problem depend on  $|\mathcal{X}|$  and  $|\text{Gen}(\mathcal{X})|$ , rather than on  $|D|$  and  $|\text{Gen}(D)|$  as in the case of the Consistency problem (12).<sup>6</sup> This generous definition of the size of the problem makes the intractability result that follows stronger. I will call (27) the *universal OT Restrictiveness problem*.

<sup>6</sup> Here,  $|\text{Gen}(\mathcal{X})|$  is defined analogously to (11), as the cardinality of the largest candidate set  $\text{Gen}(x)$  over all underlying forms  $x \in \mathcal{X}$ .

$$(i) \quad |\text{Gen}(\mathcal{X})| = \max_{x \in \mathcal{X}} |\text{Gen}(x)|$$

Letting the size of an instance of problem (27) depend on  $|\mathcal{C}|$ ,  $|\mathcal{X}|$  and  $|\text{Gen}(\mathcal{X})|$  straightforwardly ensures that the problem is in  $\mathcal{NP}$ , namely that it admits an efficient verification algorithm.

- (27) *Given:* a) typological specifications  $\mathcal{X}$ ,  $\mathcal{Y}$ ,  $Gen$  and  $\mathcal{C}$ ,  
 b) a finite OT-compatible data set  $D \subseteq \mathcal{X} \times \mathcal{Y}$ ;  
*Find:* a ranking  $\gg$  over the constraint set  $\mathcal{C}$  such that:  
 a)  $\gg$  is OT-consistent with the data set  $D$ ;  
 b) there is no ranking  $\gg'$  that satisfies (a) and such that the language corresponding to  $\gg'$  is a proper subset of the language corresponding to  $\gg$ .  
*Size:*  $\max\{|\mathcal{C}|, |\mathcal{X}|, |Gen(\mathcal{X})|\}$ .

Prince and Tesar (2004) offer an important alternative formulation of the OT Restrictiveness problem (27). They introduce the notion of a *strictness measure*, as a function  $\mu$  that takes a ranking  $\gg$  and returns a number  $\mu(\gg)$  that provides a relative measure of the size of the language  $\mathcal{L}(\text{OT}_{\gg})$  corresponding to  $\gg$ , in the sense that the (strict) monotonicity property in (28) holds for any two rankings  $\gg, \gg'$ : if the language  $\mathcal{L}(\text{OT}_{\gg'})$  corresponding to  $\gg'$  is a proper subset of the language  $\mathcal{L}(\text{OT}_{\gg})$  corresponding to  $\gg$ , then the strictness measure of  $\gg'$  is strictly smaller than the strictness measure of  $\gg$ . Thus, the smaller the strictness measure, the smaller the language.

$$(28) \quad \mathcal{L}(\text{OT}_{\gg'}) \subset \mathcal{L}(\text{OT}_{\gg}) \implies \mu(\gg') < \mu(\gg).$$

If  $\mu$  is indeed a strictness measure, then any solution of problem (29) is guaranteed to be a solution of the original Restrictiveness problem (26). In fact, if a ranking  $\gg$  is a solution of problem (29), then there cannot exist any other ranking  $\gg'$  OT-consistent with  $D$  whose corresponding language  $\mathcal{L}(\text{OT}_{\gg'})$  is a proper subset of the language  $\mathcal{L}(\text{OT}_{\gg})$  corresponding to  $\gg$ , since (28) would then imply that the strictness measure  $\mu(\gg')$  of  $\gg'$  is strictly smaller than the strictness measure  $\mu(\gg)$  of  $\gg$ , thus contradicting the hypothesis that  $\gg$  is a solution of problem (29).

- (29) *given:* a data set  $D$  of underlying/winner form pairs corresponding to some target grammar in the actual OT typology  $(\mathcal{X}^{\text{act}}, \mathcal{Y}^{\text{act}}, Gen^{\text{act}}, \mathcal{C}^{\text{act}})$ ;  
*find:* a ranking  $\gg$  over the actual constraint set  $\mathcal{C}^{\text{act}}$  such that:  
 a)  $\gg$  is OT-consistent with the data set  $D$ ,  
 b) there is no ranking  $\gg'$  that satisfies (a) and such that the strictness measure  $\mu(\gg')$  of  $\gg'$  is smaller than the strictness measure  $\mu(\gg)$  of  $\gg$ .

The reformulation (29) looks promising. Yet, we now need to come up with a strictness measure. Of course, not just any strictness measure will do. For instance, the function (30), that pairs a ranking  $\gg$  with the cardinality of the corresponding language  $\mathcal{R}(\text{OT}_{\gg})$ , trivially satisfies (28), and is thus a strictness measure. Yet, this is not a *useful* strictness measure, because there seems to be no way to compute  $\mu(\gg)$  without actually computing the corresponding language  $\mathcal{L}(\text{OT}_{\gg})$ .

$$(30) \quad \mu(\gg) = \text{cardinality of the language } \mathcal{L}(\text{OT}_{\gg}).$$

Prince and Tesar (2004) suggest a better candidate. As usual, assume that the set of constraints  $\mathcal{C} = \mathcal{F} \cup \mathcal{M}$  is split up into the subset  $\mathcal{F}$  of faithfulness constraints and the subset  $\mathcal{M}$  of markedness constraints. Consider the function  $\mu_{\text{PT}}$  in (31), that maps a ranking  $\gg$  into the number  $\mu_{\text{PT}}(\gg)$  of pairs of a faithfulness and a markedness constraint such that the former is  $\gg$ -ranked above the latter, called the *PT-measure* of the ranking  $\gg$ .

$$(31) \quad \mu_{\text{PT}}(\gg) \stackrel{\text{def}}{=} \left| \left\{ (F, M) \in \mathcal{F} \times \mathcal{M} \mid F \gg M \right\} \right|$$

Prince and Tesar (2004) reason that faithfulness (markedness) constraints work toward (against) the preservation of underlying contrasts, so that a small (large) language is likely to arise by ranking high the markedness (faithfulness) constraints. To illustrate, note that the PT-measure of the ranking in (25a) is 2 while the PT-measure of the desired ranking in (25b) is just 1. They thus conjecture that the PT-measure  $\mu_{\text{PT}}$  in (31) is indeed a strictness measure for the actual typological specifications  $(\mathcal{X}^{\text{act}}, \mathcal{Y}^{\text{act}}, Gen^{\text{act}}, \mathcal{C}^{\text{act}})$  and suggest that the original problem (26) can be replaced with problem (29) using the PT-measure (31) as a strictness measure. As usual, lack of firm knowledge of the actual typological specifications  $(\mathcal{X}^{\text{act}}, \mathcal{Y}^{\text{act}}, Gen^{\text{act}}, \mathcal{C}^{\text{act}})$  prompts us to switch to the universal formulation of the problem in (32): given a data set  $D$  corresponding to a language in a known OT typology, we have to find a ranking that minimizes Prince and Tesar's alleged strictness measure among those consistent with the data  $D$ . Here, I have complemented the statement of the problem with the specification of the size of its instances. The core idea of

strictness measures is to be able to determine the relative restrictiveness of two rankings without computing the entire corresponding languages. Thus it makes sense to let the size of an instance of problem (32) depend only on the cardinality  $|D|$  of the data set and on the cardinality  $|Gen(D)|$  of the largest candidate set among the underlying forms in  $D$ , rather than on the cardinality  $|\mathcal{X}|$  of the entire set of forms and on the cardinality  $|Gen(\mathcal{X})|$  of the largest candidate set among all underlying forms in  $\mathcal{X}$ , as I did above for the case of the original problem (27). Furthermore, let me point out once more that letting the size of an instance of the problem depend on the cardinality  $|Gen(D)|$  of the largest candidate set means that a solution algorithm can afford the time to list and inspect all candidates. Again, this generous definition of the size makes the intractability result that follows stronger.<sup>7</sup> I will call problem (32) the *(universal) OT PT-Restrictiveness problem*.<sup>8</sup>

- (32) *Given:* a) typological specifications  $(\mathcal{X}, \mathcal{Y}, Gen, \mathcal{C})$ ,  
 b) a finite OT-consistent data set  $D \subseteq \mathcal{X} \times \mathcal{Y}$ .

*Find:* a ranking  $\gg$  over the constraint set  $\mathcal{C}$  such that:

- a)  $\gg$  is OT-consistent with the data set  $D$ ;  
 b) there is no ranking  $\gg'$  that satisfies (a) such that the PT-strictness measure  $\mu_{PT}(\gg')$  of  $\gg'$  is smaller than the PT-strictness measure  $\mu_{PT}(\gg)$  of  $\gg$ .

*Size:*  $\max\{|D|, |\mathcal{C}|, |Gen(D)|\}$ .

The main result of this paper is the following claim 2. As recalled in section 2, the switch from the *actual* to the *universal* formulation is harmless for the case of the Consistency problem: Tesar and Smolensky's (1998) claim 1 guarantees that the complexity class of the two formulations of the problem is the same, as they are both "easy". Claim 2 says that the situation is very different for the case of the Restrictiveness problem, as the latter problem cannot be solved in its universal formulation. As elaborated in section 4, specific assumptions on the underlying OT typologies are needed, in order to introduce further structure into the problem that can be exploited by solution algorithms.

**Claim 2.** *The problem of the acquisition of phonotactics in OT cannot be solved (it is NP-complete), both in its original formulation as the (universal) OT Restrictiveness problem (27) as well as in Prince and Tesar's reformulation (32).*

A proof of claim 2 is provided in the final Appendix. Here is an outline of the reasoning. Given an arbitrary finite set  $A = \{a, b, \dots\}$  with cardinality  $|A|$ , consider a set  $T$  of triplets of elements of  $A$ . The set  $T$  is called *linearly cyclically compatible* iff there exists a one-to-one function  $\pi : A \rightarrow \{1, 2, \dots, |A|\}$  such that for every triplet  $(a, b, c) \in T$  either  $\pi(a) < \pi(b) < \pi(c)$  or  $\pi(b) < \pi(c) < \pi(a)$  or  $\pi(c) < \pi(a) < \pi(b)$ . This notion is illustrated in (33): the set  $T$  in (33a) is linearly cyclically compatible; the one in (33b) is not.

- (33)  $A = \{a, b, c, d\}$   
 a.  $T = \{(a, b, c), (b, c, d)\}$   
 b.  $T = \{(a, b, c), (a, c, b)\}$

Consider the *Cyclic Ordering problem* in (34). Galil and Megiddo (1977) prove that the problem cannot be solved efficiently (it is NP-complete). I show that the PT-Restrictiveness problem (32) cannot be solved efficiently (it is NP-complete), as any algorithm that would solve the latter problem efficiently can be used to construct an algorithm that solves efficiently the Cyclic Ordering problem. Finally, I deduce from the fact that the PT-Restrictiveness problem is not solvable, that also the original formulation (27) of the Restrictiveness problem is not solvable either, thus completing the proof of claim 2.

- (34) *Given:* a) a finite set  $A$ ;  
 b) a collection  $T \subseteq A \times A \times A$  of triplets of elements of  $A$ ;  
*Find:* "yes" iff  $T$  is linearly cyclically compatible;  
*Size:* the cardinality  $|A|$  of  $A$ .

<sup>7</sup> Letting the size of an instance of problem (32) depend not only on  $|\mathcal{C}|$  and  $|D|$ , but also on  $|Gen(D)|$  straightforwardly ensures that the problem is in  $\mathcal{NP}$ , namely that it admits an efficient verification algorithm.

<sup>8</sup>The Consistency problem (9) corresponds to *Empirical Risk Minimization* in the Statistical Learning literature, while problem (32) corresponds to a *regularized* version thereof, with regularization function  $\mu$ .

The proof actually shows that the OT Restrictiveness problem (27) is intractable even when a solution algorithm is provided with the time to list all candidates. In other words, this hardness result is orthogonal to other hardness results for OT available in the literature; see Eisner (1997), Idsardi (2006), Wareham (1998), and Heinz et al. (2009) for discussion. Furthermore, I show that the Restrictiveness problem remains intractable even when restricted to data with the simplest “disjunctive structure”, in the sense that for each underlying/winner/loser form triplet there are at most two winner-preferrers.<sup>9</sup>

#### 4. CONCLUSIONS AND FUTURE DIRECTIONS

A plausible conjecture is that the actual learning strategies adopted by humans in acquiring their mother language have been selected by evolution based on considerations of computational soundness and efficiency. Thus, purely *computational* considerations might lead to *cognitively* plausible algorithmic models of language acquisition. It is from this perspective that computational considerations gain currency within cognitive sciences. The resulting approach could well be called *Cognitive Computational Linguistics*, as it uses computational tools to derive cognitively plausible models. The research reported in this paper fits squarely within this framework. Here, I have focused on a specific sub-problem of the overall task of language acquisition, namely the problem of the acquisition of phonotactics. The fact that phonotactics is acquired (at least in part) at a relatively early age, prior to the development of other linguistic knowledge, suggests that the problem of the acquisition of phonotactics can be tackled in isolation; see Hayes (2004) for discussion. I have thus singled out this specific sub-problem, roughly stated as the Restrictiveness problem (2), repeated in (35). Following Heinz et al. (2009), I have considered the *universal* formulation of the problem, whereby the typology varies arbitrarily as an input to the problem. I have provided an explicit, formal statement of the problem within the framework of OT and I have shown that the problem cannot be solved efficiently. This hardness result brings out the intrinsic difficulty of the learning task, as the problem remains hard even if we can afford the time to list and inspect all candidates.

- (35) *Given:* a) a typology,  
           b) a finite set of data drawn from a language in the typology;  
*Find:* a grammar in the typology such that:  
           a) its phonotactics is *consistent* with the data;  
           b) its corresponding language is as *restrictive* as possible (w.r.t. set inclusion).

How should this hardness result be interpreted? what are its consequences for the development of proper models of the acquisition of phonotactics? To start, I conjecture that this complexity result has nothing to do with the choice of the OT framework, namely that an analogous result holds for the corresponding problem within alternative frameworks, such as Harmonic Grammar; see Legendre et al. (1990b,a). As a matter of fact, the intuition that Restrictiveness is though to achieve dates back to the very beginning of Generative Linguistics, as already informally conjectured in Manzini and Wexler (1987). The advantage of OT is that its formally explicit nature makes it possible to state and prove explicit complexity results. Following Heinz et al. (2009), I conjecture that what makes the problem (35) hard is not the choice of the typological framework rather the fact that no restrictions are placed on the possible typologies, because of the universal formulation of the problem. The comparison with the Consistency problem here is instructive. Recall that Tesar and Smolensky’s (1998) claim 1 ensures that the Consistency problem is solvable even in its universal formulation. In other words, the ranking logic of OT contains enough structure to support efficient solution algorithms. The Restrictiveness problem is more demanding, and thus the bare structure provided by OT’s ranking logic is not enough. Further structure needs to be introduced into the problem in the form of explicit restrictions on typological specifications (i.e. on generating functions and constraint sets) in order to support efficient solution algorithms. Thus, the significance of the complexity result provided by claim 2 of this paper is to solidly shape the future research on the problem of the acquisition of phonotactics around the core question (36). Question (36) paradigmatically illustrates the field of Cognitive Computational Linguistics, as it interfaces directly the modeling perspective (through the requirement of phonological plausibility) with the computational one (through the tractability requirement).

<sup>9</sup>Of course, the Restrictiveness problem is trivial for data sets that have a unique winner-preferrer per underlying/winner/loser form triplet, as those data are OT-consistent with a unique ranking, and thus the Restrictiveness problem coincides with the Consistency problem in this case.

- (36) Individuate phonologically plausible assumptions on OT typological specifications (namely assumptions on generating function and constraint set) that provably make the problem of the acquisition of phonotactics tractable.

Let me illustrate question (36), by sketching a possible way to address it. One of the goals of *segmental phonotactics* is to model knowledge of the inventory of licit segments in a given language as well as of the licit segment concatenations. Here is a phonologically plausible formal framework for segmental phonotactics. Segments are described through  $N$  partial, binary phonological features  $\varphi_1, \dots, \varphi_N$ , as in (37a). And we are interested, say, in licit concatenations of two segments, as in (37b). Assume that the generating function is allowed to change arbitrarily the feature values. Furthermore, it is allowed to delete one of the two segments in the concatenation (37b). The constraint set might contain faithfulness constraints against deletion and epenthesis. Furthermore, each feature might come with a dedicated *identity faithfulness constraint* and a dedicated *markedness constraint*. Finally, besides these *unary* constraints that target a single feature, there is a set of markedness constraints of *higher arity*, that punish certain combinations of values of certain sets of features, either within the same segment (37a) or across two segments (37b). These markedness constraints of higher arity thus introduce *feature interaction* into the model. A number of constraints in segmental phonotactics can be fit into this framework; see for instance Lombardi (1999), Beckman (1997), Steriade (1999), Jun (2004), Kaun (1995, Chp. 6), etcetera.<sup>10</sup> To start, assume that the markedness constraints of higher arity can target at most two features. These binary markedness constraints define a *markedness graph* over the feature set: two features are connected by an edge in the graph provided that the two features interact through a binary markedness constraint. Within the framework just sketched, the general question (36) can be restated more precisely as follows: which assumptions on the markedness graph are needed to make the problem of the acquisition of phonotactics (35) tractable? and which assumptions are needed on the patterns of feature values punished by the binary markedness constraints? And how phonologically plausible are these restrictive assumptions?

$$(37) \quad \text{a.} \quad \begin{bmatrix} \varphi_1 \\ \vdots \\ \varphi_N \end{bmatrix} \quad \text{b.} \quad \begin{bmatrix} \varphi_1 \\ \vdots \\ \varphi_N \end{bmatrix} \begin{bmatrix} \varphi'_1 \\ \vdots \\ \varphi'_N \end{bmatrix}$$

Let me close by pointing out some alternative strategies to cope with the intractability of the problem (35) of the acquisition of phonotactics, besides the strategy of restricting the formulation of the problem to special families of OT typologies, as outlined in (36). One alternative approach is to investigate how the complexity of the problem is affected by fixed constraint rankings, such as fixed *markedness hierarchies* and Steriade's (2001) *P-map* for faithfulness constraints.<sup>11</sup> Yet another strategy is to resort to an *approximate* solution of problem (35), settling on a "small" language rather than a smallest language. The theory of *approximation algorithms* is a recent and fast growing branch of Computer Science and Integer Programming; see for instance Bertsimas and Weismantel (2005). Here is a concrete way to implement this strategy. In Magri (2007, 2008), I note that Prince and Tesar's formulation (32) of the problem of the acquisition of phonotactics can be straightforwardly restated as an *Assignment problem with linear side constraints* (henceforth: AssignLSCPbm). The core idea behind this observation is that OT rankings can be represented as permutation matrices, that the PT-measure (31) can be described as a linear function over matrices and that OT-consistency with a data set  $D$  can be enforced through a collection of linear inequalities over permutation matrices. The AssignLSCPbm is a classical computational problem, for which powerful approximation algorithms exist; see for instance Arora et al. (2002). Representing the universal PT-Restrictiveness problem as an AssignLSCPbm paves the way to the application of existing approximation algorithms for the AssignLSCPbm to the OT problem of the acquisition of phonotactics.

## 5. APPENDIX

Subsection 5.2 presents a proof of claim 2 that the problem of the acquisition of phonotactics is intractable. As anticipated at the end of section 3, the proof is a standard complexity argument,

<sup>10</sup>Thanks to Donca Steriade (p.c.) for discussion on this point.

<sup>11</sup>Thanks to Adam Albright (p.c.) for pointing out this possibility to me.

by reduction from the well known NP-complete Cyclic Ordering problem (34). The interested reader can find some standard preliminaries on Complexity Theory in subsection 5.1.

**5.1. Preliminaries on Complexity Theory.** Given two sets  $\mathcal{I}$  and  $\mathcal{S}$ , a corresponding *problem* is any relation  $\Pi \subseteq \mathcal{I} \times \mathcal{S}$  between  $\mathcal{I}$  and  $\mathcal{S}$ . Any element in the set  $\mathcal{I}$  is called an *instance* of the problem; every element in the set  $\mathcal{S}$  is called a *SOLUTION* of the problem. A problem  $\Pi \subseteq \mathcal{I} \times \mathcal{S}$  can also be represented as in (38). The Consistency problem (10), the Restrictiveness problem (26), and the PT-Restrictiveness problem (26) all illustrate the general scheme (38).

- (38) *Given:* an instance  $x \in \mathcal{I}$ ;  
*Find:* a solution  $y \in \mathcal{S}$  such that  $\Pi(x, y)$ .

Assume that the set  $\mathcal{I}$  of instances comes with a function  $|\cdot| : \mathcal{I} \rightarrow \mathbb{N}$  that pairs each instance  $x$  of the problem  $\Pi$  with a number  $|x|$  that expresses the *size* of that instance and captures its complexity relative to other instances of the problem. A problem  $\Pi = (\Pi, |\cdot|)$  is called *tractable* provided it admits an efficient *solution algorithm*, namely an algorithm  $\text{Solve}_\Pi$  that satisfies condition (39).

- (39) For every instance  $x \in \mathcal{I}$ ,  $\text{Solve}_\Pi$  runs on input  $x$  in time polynomial in its size  $|x|$  and returns a solution  $y = \text{Solve}_\Pi(x)$  such that  $\Pi(x, y)$ .

At least in principle, it is straightforward to show that a given problem is tractable: one just needs to exhibit an efficient solution algorithm, in the sense of (39). On the contrary, it is not at all trivial to show that a given problem is *not* tractable. Naïvely, one would have to show that no polynomial-time solution algorithm exists. This strategy is of course not viable. An alternative, more sophisticated strategy has been devised within Complexity Theory. This alternative strategy is informally stated in (40). The rationale is as follows: if there exist non-tractable problems and if  $\Pi'$  is among the hardest problems, then  $\Pi'$  must be non-tractable; thus, if our problem  $\Pi$  is at least as hard as  $\Pi'$ , then our problem  $\Pi$  has got to be non-tractable too.

- (40) Suppose that non-tractable problems exist. To conclude that a given problem  $\Pi$  is non-tractable, show that  $\Pi$  is at least as hard as some other problem  $\Pi'$  that is known to be among the hardest problems.

In the rest of this section, I review the classical formalization of the intuition (40), one step at the time; see Garey and Johnson (1979) and Cormen et al. (1990, Ch. 36) for details.

**5.1.1. First step.** A *decision problem* is a problem  $\Pi \subseteq \mathcal{I} \times \mathcal{S}$  whose set of solutions is  $\mathcal{S} = \{\text{yes}, \text{no}\}$ . Intuitively, decision problems are those problems that only ask for a “yes” or a “no”. For instance, the following problem (41) is the decision problem corresponding to Prince and Tesar’s problem (32), in the sense that non-tractability of the decision problem (41) entails non-tractability of the original problem (32). In fact, if the original problem (32) were tractable, then the decision problem (41) would be tractable too, since I could solve the decision problem by finding a solution  $\gg$  of the original problem (32) and then checking whether  $\mu_{\text{PT}}(\gg)$  is smaller than the threshold  $k$  or not.

- (41) *Given:* a) universal specifications  $\mathcal{X}, \mathcal{Y}, \text{Gen}$  and  $\mathcal{C}$ ;  
b) a finite OT-consistent data set  $D \subseteq \mathcal{X} \times \mathcal{Y}$ ;  
c) an threshold  $k$ ;  
*Find:* “yes” iff there exists a ranking  $\gg$  OT-consistent with the data set  $D$  s.t. its PT-measure  $\mu_{\text{PT}}(\gg)$  is at most  $k$ .

The preceding considerations extend to the general case. The decision variant  $\Pi_{\text{dec}}$  of a problem  $\Pi$  is obtained by introducing a threshold  $k$  and then asking the question of whether there exists a solution that satisfies the desired property up to the threshold  $k$ . It is in general the case that non-tractability of the decision problem  $\Pi_{\text{dec}}$  entails non-tractability of the original problem  $\Pi$ . Thus, the initial informal statement (40) can be restated as in (42).

- (42) Suppose that non-tractable decision problems exist. To conclude that a given problem  $\Pi$  is non-tractable, show that the corresponding decision problem  $\Pi_{\text{dec}}$  is at least as hard as some other decision problem  $\Pi'$  that is known to be among the hardest decision problems.

Note that (42) is now fully stated in terms of decision problems. The next step towards a proper formalization of (42) consists of making explicit the condition that the decision problem  $\Pi_{\text{dec}}$  is at least as hard as some other decision problem  $\Pi'$ .

5.1.2. *Second step.* Given two decision problems  $\Pi_1 \subseteq \mathcal{J}_1 \times \mathcal{S}_1$  and  $\Pi_2 \subseteq \mathcal{J}_2 \times \mathcal{S}_2$ , we say that  $\Pi_1$  is *not harder than* or that it *reduces to*  $\Pi_2$  (in symbols:  $\Pi_1 \leq_P \Pi_2$ ) iff there exists an algorithm  $\text{Reduction}_{\Pi_1, \Pi_2}$  that satisfies condition (43). In this case, the algorithm  $\text{Reduction}_{\Pi_1, \Pi_2}$  is called a *reduction* of  $\Pi_1$  to  $\Pi_2$ . The relation  $\leq_P$  is reflexive, anti-symmetric and transitive, namely it is a partial order among decision problems. The two problems  $\Pi_1$  and  $\Pi_2$  are called *equivalent* iff both  $\Pi_1 \leq_P \Pi_2$  and  $\Pi_2 \leq_P \Pi_1$ .

- (43) For every instance  $x_1 \in \mathcal{J}_1$  of problem  $\Pi_1$ ,  $\text{Reduction}_{\Pi_1, \Pi_2}$  runs on  $x_1$  in time polynomial in its size  $|x_1|$  and returns an instance  $x_2 = \text{Reduction}_{\Pi_1, \Pi_2}(x_1) \in \mathcal{J}_2$  of problem  $\Pi_2$  such that  $|x_2|$  is polynomial in  $|x_1|$  and furthermore  $\Pi_1(x_1) = \text{yes}$  iff  $\Pi_2(x_2) = \text{yes}$

In other words, problem  $\Pi_1$  reduces to problem  $\Pi_2$  provided that any efficient solution algorithm  $\text{Solve}_{\Pi_2}$  for  $\Pi_2$  can be turned into the efficient solution algorithm  $\text{Solve}_{\Pi_1}$  for  $\Pi_1$  described in (44). In other words, the issue of solving problem  $\Pi_1$  can be reduced to the issue of solving problem  $\Pi_2$  and problem  $\Pi_1$  thus cannot be harder than problem  $\Pi_2$ .

- (44)  $\text{Solve}_{\Pi_1}(x_1)$   
 1 compute  $x_2 = \text{Reduction}_{\Pi_1, \Pi_2}(x_1)$   
 2 return  $\text{Solve}_{\Pi_2}(x_2)$

In conclusion, in order to show that a given decision problem  $\Pi_2$  is not tractable, it is sufficient to show that there is a decision problem  $\Pi_1$  such that  $\Pi_1 \leq_P \Pi_2$  and furthermore  $\Pi_1$  is not tractable. I thus further formalize (42) as in (45).

- (45) Suppose that non-tractable decision problems exist. To conclude that a given problem  $\Pi$  is non-tractable, show that  $\Pi' \leq_P \Pi_{\text{dec}}$ , where  $\Pi_{\text{dec}}$  is the decision problem corresponding to  $\Pi$  and  $\Pi'$  is some decision problem that is known to be among the hardest decision problems.

The next step towards a proper formalization of (45) consists of making explicit the assumption that there exist non-tractable decision problems.

5.1.3. *Third step.* The set of tractable decision problems is denoted by  $\mathcal{P}$ . More explicitly, a decision problem  $\Pi = (\Pi, |\cdot|)$  belongs to the class  $\mathcal{P}$  iff it admits an algorithm  $\text{Solve}_{\Pi}$  that satisfies condition (46). This condition (46) is a straightforward adaptation to the case of decision problems of the general condition (39).

- (46) For every instance  $x \in \mathcal{J}$ ,  $\text{Solve}_{\Pi}$  runs on input  $x$  in time polynomial in its size  $|x|$  and furthermore  $\Pi(x) = \text{yes}$  iff  $\text{Solve}_{\Pi}(x) = \text{yes}$

Another important class of decision problems is  $\mathcal{NP}$ : a decision problem  $\Pi$  belongs to the class  $\mathcal{NP}$  iff  $\Pi$  admits a polynomial-time *verification algorithm*, namely an algorithm  $\text{Verify}_{\Pi}$  that satisfies condition (47) for some polynomial  $p$ . For instance, the decision problem (41) belongs to the class  $\mathcal{NP}$ : given a ranking (encoded as a not too long boolean vector  $y$ ), it is easy to decide whether its corresponding PT-measure (31) is smaller than  $k$ .

- (47) For every instance  $x \in \mathcal{J}$ ,  $\Pi(x) = \text{yes}$  iff there exists  $y \in \{0, 1\}^{p(|x|)}$  such that  $\text{Verify}_{\Pi}$  runs on input  $(x, y)$  in time polynomial in the size  $|x|$  and returns “yes”.

Given an arbitrary problem  $\Pi \in \mathcal{NP}$ , we can use the corresponding verification algorithm  $\text{Verify}_{\Pi}$  to construct the algorithm  $\text{Solve}_{\Pi}$  in (48). Of course,  $\text{Solve}_{\Pi}$  is a solution algorithm for  $\Pi$  with worst-case running time of the order of  $2^{p(|x|)}$ . Thus,  $\mathcal{NP}$  is the class of decision problems for which brute force search yields an exponential time solution algorithm.

- (48)  $\text{Solve}_{\Pi}(x)$   
 1 *answer*  $\leftarrow$  no  
 2 for every  $y \in \{0, 1\}^{p(|x|)}$   
 3     if  $\text{Verify}_{\Pi}(x, y) = \text{yes}$  then  
 4         *answer*  $\leftarrow$  yes  
 5 return *answer*

Of course,  $\mathcal{P} \subseteq \mathcal{NP}$ , since any polynomial-time solution algorithm  $\text{Solve}_{\Pi}$  for  $\Pi$  can be used as a verification algorithm. Do the two classes  $\mathcal{P}$  and  $\mathcal{NP}$  coincide?; namely: do all decision problems that admit an exponential time solution algorithm also admit a polynomial time solution algorithm? This question is currently open in the literature. The *complexity conjecture* (49) says that there are indeed problems in  $\mathcal{NP}$  that are not tractable and thus do not belong to  $\mathcal{P}$ , in the sense that they do not admit an efficient solution algorithm.

$$(49) \quad \mathcal{P} \neq \mathcal{NP}$$

The complexity conjecture (49) formalizes the crucial assumption in (45) that there exist non-tractable decision problems. The statement in (45) can thus be formalized as in (50).

$$(50) \quad \text{Suppose that } \mathcal{P} \neq \mathcal{NP}. \text{ To conclude that a given problem } \Pi \text{ is non-tractable, show that } \Pi' \leq_{\mathcal{P}} \Pi_{\text{dec}}, \text{ where } \Pi_{\text{dec}} \text{ is the decision problem corresponding to } \Pi \text{ and } \Pi' \text{ is some decision problem that is known to be among the hardest decision problems.}$$

The last step towards a proper formalization of (50) consists of making explicit the assumption that  $\Pi'$  is among the hardest decision problems.

5.1.4. *Fourth step.* A decision problem  $\Pi$  is called *hard* iff the following condition (51) holds. This definition says that NP-hard problems are at least as hard as any problem in  $\mathcal{NP}$ . In other words, it says that, if we had a polynomial-time solution algorithm for even just one NP-hard problem, then we would have a polynomial-time solution algorithm for every problem in  $\mathcal{NP}$ . A decision problem is called *NP-complete* iff it is hard and furthermore it belongs to the class  $\mathcal{NP}$ .

$$(51) \quad \Pi' \leq_{\mathcal{P}} \Pi \text{ for every decision problem } \Pi' \in \mathcal{NP}.$$

NP-complete problems are thus among the hardest decision problems. I can thus conclude this subsection with the fully explicit restatement of (50) provided in (52).

$$(52) \quad \text{Suppose that } \mathcal{P} \neq \mathcal{NP}. \text{ To conclude that a given problem } \Pi \text{ is non-tractable, show that } \Pi' \leq_{\mathcal{P}} \Pi_{\text{dec}}, \text{ where } \Pi_{\text{dec}} \text{ is the decision problem corresponding to } \Pi \text{ and } \Pi' \text{ is some NP-complete decision problem.}$$

In the next subsection, I will use (52) to show that the acquisition of phonotactics is intractable, both in its original formulation as the (universal) OT Restrictiveness problem (27) as well as in Prince and Tesar's reformulation (32).

5.2. **Proof of claim 2.** Following a large literature, in section 3, I have formalized the problem of the acquisition of phonotactics in OT as the universal Restrictiveness problem (27), repeated in (53). We are provided with an OT typology and with some data drawn from a language in that typology, and we have to find in the typology an OT grammar consistent with the data that corresponds to a smallest language. In order to focus on the restrictiveness task, we are allowed the time to list and inspect all candidates (as the size of an instance depends on the cardinality  $|\text{Gen}(\mathcal{X})|$  of the largest candidate set).

$$(53) \quad \begin{array}{ll} \textit{Given:} & \text{a) typological specifications } (\mathcal{X}, \mathcal{Y}, \textit{Gen}, \mathcal{C}), \\ & \text{b) a finite OT-consistent data set } D \subseteq \mathcal{X} \times \mathcal{Y}; \\ \textit{Find:} & \text{a ranking } \gg \text{ over the constraint set } \mathcal{C} \text{ such that:} \\ & \text{a) } \gg \text{ is OT-consistent with the data set } D; \\ & \text{b) there is no ranking } \gg' \text{ that satisfies (a) and such that the language corresponding to } \gg' \text{ is a proper subset of the language corresponding to } \gg. \\ \textit{Size:} & \max \{|\mathcal{C}|, |\mathcal{X}|, |\textit{Gen}(\mathcal{X})|\}. \end{array}$$

The decision problem corresponding to (53) is provided in (54). As noted for the general case in 5.1.1, intractability of the decision problem (54) entails intractability of the original problem (53). In fact, if problem (53) can be solved, then the decision problem (54) can be solved too: given an instance of the latter, find a solution  $\gg$  of the corresponding instance of the former and then just check whether the cardinality of the corresponding language is at most  $k$ .<sup>12</sup> From now on, I will refer to the both the original problem (53) and the decision problem (54) as the (*universal*) *OT Restrictiveness problem*.

<sup>12</sup>Note that the generous dependence of the size of an instance of problem (53) on  $|\mathcal{X}|$  and  $|\textit{Gen}(\mathcal{X})|$  provides sufficient time to trivially compute the language corresponding to a given ranking.

- (54) *Given:* a) typological specifications  $(\mathcal{X}, \mathcal{Y}, Gen, \mathcal{C})$ ,  
 b) a finite OT-consistent data set  $D \subseteq \mathcal{X} \times \mathcal{Y}$ ,  
 c) a threshold  $k$ ;  
*Output:* “yes” iff there is a ranking  $\gg$  OT-consistent with  $D$  s.t. the corresponding language has cardinality at most  $k$ ;  
*Size:*  $\max\{|\mathcal{C}|, |\mathcal{X}|, |Gen(\mathcal{X})|\}$ .

In section 3, I have also considered Prince and Tesar’s (2004) formulation (32) of the problem of the acquisition of phonotactics, repeated in (55). Recall that the idea of this formulation is to measure the size of the language corresponding to a ranking  $\gg$  in terms of a strictness measure  $\mu(\gg)$ . The specific strictness measure considered here is the PT-measure  $\mu_{PT}(\gg)$  in (31), that counts the number of pairs of a faithfulness constraint  $\gg$ -ranked above a markedness constraint, encoding the intuition that small languages arise from low ranking the faithfulness constraints.

- (55) *Given:* a) typological specifications  $(\mathcal{X}, \mathcal{Y}, Gen, \mathcal{C})$ ,  
 b) a finite OT-consistent data set  $D \subseteq \mathcal{X} \times \mathcal{Y}$ .  
*Find:* a ranking  $\gg$  over the constraint set  $\mathcal{C}$  such that:  
 a)  $\gg$  is OT-consistent with the data set  $D$ ;  
 b) there is no ranking  $\gg'$  that satisfies (a) such that the PT-strictness measure  $\mu_{PT}(\gg')$  of  $\gg'$  is smaller than the PT-strictness measure  $\mu_{PT}(\gg)$  of  $\gg$ .  
*Size:*  $\max\{|D|, |\mathcal{C}|, |Gen(D)|\}$ .

Problem (55) can be restated in comparative notation as in (56), in the sense that a ranking solves an instance of problem (55) corresponding to a data set  $D$  iff it solves the instance of problem (56) corresponding to the comparative tableau  $\mathbf{A} = \mathbf{A}_D$  corresponding to the data set  $D$ , as defined in (18). This equivalence crucially depends on the assumption that the size of an instance of problem (55) generously depends not only on  $|\mathcal{C}|$  and on  $|D|$ , but also on  $|Gen(D)|$ , which allows the size of an instance of the corresponding problem (56) to depend on the number  $m$  of rows of the input tableau. The set  $\mathcal{F}$  provided with an instance of problem (56) just encodes which one of the  $n$  columns of the comparative tableau  $\mathbf{A}$  correspond to faithfulness constraints, so that the PT-measure  $\mu_{PT}$  is well defined.

- (56) *Given:* a) an OT-consistent comparative tableau  $\mathbf{A}$  with  $m$  rows and  $n$  columns,  
 b) the set  $\mathcal{F} \subseteq \{1, \dots, n\}$  of columns corresponding to faithfulness constraints;  
*Find:* a ranking  $\gg$  such that:  
 a)  $\gg$  is OT-consistent with the comparative tableau;  
 b) there is no ranking  $\gg'$  that satisfies (a) such that the PT-strictness measure  $\mu_{PT}(\gg')$  of  $\gg'$  is smaller than the PT-strictness measure  $\mu_{PT}(\gg)$  of  $\gg$ .  
*Size:*  $\max\{m, n\}$

The decision problem corresponding to (56) is provided in (57). Once again, intractability of the decision problem (57) entails intractability of the original problem (56). In fact, if the original problem (56) can be solved in polynomial time, then the corresponding decision problem (57) can be solved in polynomial time too: given an instance of the decision problem (57), find a solution  $\gg$  of the corresponding instance of (56) and then just check whether  $\gg$  has PT-strictness measure  $\mu_{PT}(\gg)$  that is at most  $k$ . From now on, I will refer to the both the original problem (56) and the decision problem (57) as the *(universal) PT-Restrictiveness problem*.

- (57) *Given:* a) an OT-consistent comparative tableau  $\mathbf{A}$  with  $m$  rows and  $n$  columns,  
 b) the set  $\mathcal{F} \subseteq \{1, \dots, n\}$  of columns corresponding to faithfulness constraints;  
 c) an threshold  $k$ ;  
*Output:* “yes” iff there is a ranking  $\gg$  OT-consistent with  $\mathbf{A}$  with PT-strictness measure  $\mu_{PT}(\gg)$  not larger than  $k$ ;  
*Size:*  $\max\{m, n\}$ .

My proof of the intractability of the decision problems (54) and (57) will make leverage on the following classical result from Complexity Theory. Given an arbitrary finite set  $A = \{a, b, c, \dots\}$  with cardinality  $|A|$ , consider a set  $T$  of triplets of elements of  $A$ . The set  $T$  is called *linearly cyclically compatible* iff there exists a one-to-one function  $\pi : A \rightarrow \{1, 2, \dots, |A|\}$  such that for

every triplet  $(a, b, c) \in T$  either  $\pi(a) < \pi(b) < \pi(c)$  or  $\pi(b) < \pi(c) < \pi(a)$  or  $\pi(c) < \pi(a) < \pi(b)$ . This notion is illustrated in (58): the set  $T$  in (58a) is linearly cyclically compatible; the one in (58b) is not.

$$(58) \quad \begin{aligned} A &= \{a, b, c, d\} \\ \text{a. } T &= \{(a, b, c), (b, c, d)\} \\ \text{b. } T &= \{(a, b, c), (a, c, b)\} \end{aligned}$$

Consider the *Cyclic Ordering problem* in (59). Galil and Megiddo (1977) prove NP-completeness of this problem by reduction from the *3-Satisfiability* problem. The Cyclic Ordering problem is listed as problem [MS2] in (Garey and Johnson, 1979, p. 279).

$$(59) \quad \begin{aligned} \text{Given:} & \text{ a) a finite set } A; \\ & \text{ b) a set } T \text{ of triplets of elements of } A; \\ \text{Output:} & \text{ "yes" iff } T \text{ is linearly cyclically compatible;} \\ \text{size:} & |A|. \end{aligned}$$

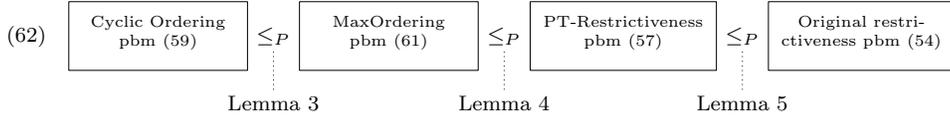
To simplify the presentation, it is useful to introduce a new, auxiliary problem. Given an arbitrary finite set  $A = \{a, b, c, \dots\}$  with cardinality  $|A|$ , consider a set  $S$  of pairs of elements of  $A$ . The set  $S$  is called *linearly compatible* iff there exists a one-to-one function  $\pi : A \rightarrow \{1, 2, \dots, |A|\}$  such that for every pair  $(a, b) \in S$  we have  $\pi(a) < \pi(b)$ . This notion is illustrated in (60): the set  $S$  in (60a) is linearly compatible; the one in (60b) is not.

$$(60) \quad \begin{aligned} A &= \{a, b, c, d\} \\ \text{a. } S &= \{(a, b), (b, c), (c, d)\} \\ \text{b. } S &= \{(a, b), (b, c), (c, a)\} \end{aligned}$$

It is useful to let  $S$  be not just a set but a *multi-set*, namely to allow for the possibility that  $S$  contains multiple instances of the same pair. The notion of cardinality and the subset relation are trivially extended from sets to multi-sets. Thus, consider the decision problem (61), that I will call the *MaxOrdering problem*.<sup>13</sup>

$$(61) \quad \begin{aligned} \text{Given:} & \text{ a) a finite set } A, \\ & \text{ b) a multi-set } P \text{ of pairs of elements of } A, \\ & \text{ c) a threshold } k \leq |P|; \\ \text{Output:} & \text{ "yes" iff there is a linearly compatible multi-set } S \subseteq P \text{ with } |S| \geq k; \\ \text{Size:} & \max\{|A|, |P|\}. \end{aligned}$$

The structure of the proof of claim 2 presented here is summarized in (62). Lemma 3 shows that the Cyclic Ordering problem can be reduced to the MaxOrdering problem. The known intractability of the Cyclic Ordering problem thus entails intractability of the MaxOrdering problem. Lemma 4 (which is the main result of the paper) then shows that the MaxOrdering problem can be reduced to the PT-Restrictiveness problem (57). The intractability of the MaxOrdering problem thus entails the intractability of Prince and Tesar's formulation (55) of the problem of the acquisition of phonotactics. Finally, Lemma 5 shows that the PT-Restrictiveness problem can be reduced to the original Restrictiveness problem (54). The intractability of the PT-Restrictiveness problem thus entails the intractability of the original formulation (53) of the problem of the acquisition of phonotactics. As all the problems considered here are obviously in  $\mathcal{NP}$ ,<sup>14</sup> their NP-completeness follows straightforwardly, concluding the proof of claim 2.



<sup>13</sup>It makes sense to let the size of an instance of the Cyclic Ordering problem (59) be just the cardinality of the set  $A$ . In fact, the cardinality of the set  $T$  of triplets of elements of  $A$  can be at most  $|A|^3$ . On the other hand, it makes sense to let the size of an instance of the MaxOrdering problem (61) depend also on the cardinality of the multi-set  $P$  of pairs of elements of  $A$  rather than only on the cardinality of the set  $A$ , as  $P$  is a *multi-set* and thus its cardinality cannot be bound in terms of the cardinality of  $A$ .

<sup>14</sup>See footnotes 6 and 7.

**Lemma 3.** *The CyclicOrdering problem (59) is reducible to the MaxOrdering problem (61).<sup>15</sup>*

*Proof.* Given an instance  $(A, T)$  of the Cyclic Ordering problem (59), consider the corresponding instance  $(A, P, k)$  of the MaxOrdering problem (61) defined as in (63). For every triplet  $(a, b, c)$  in the set  $T$ , we put in the multi-set  $P$  the three pairs  $(a, b)$ ,  $(b, c)$  and  $(c, a)$ . Furthermore, we set the threshold  $k$  to twice the number of triplets in the set  $T$ . Note that  $P$  is a multi-set because it might contain two instances of the same pair coming from two different triplets in  $T$ .

$$(63) \quad P = \left\{ (a, b), (b, c), (c, a) \mid (a, b, c) \in T \right\}$$

$$k = 2|T|$$

The construction is illustrated in (64): given the instance of CyclicOrdering in (64a), we construct the corresponding instance of MaxOrdering in (64b). Note that  $P$  is a multi-set because it contains two instances of the pair  $(b, c)$ , coming from two different triplets in  $T$ .

$$(64) \quad \text{a. } \left( A = \{a, b, c, d\}, \quad T = \left\{ \begin{array}{l} (a, b, c), \\ (b, c, d) \end{array} \right\} \right)$$

$$\text{b. } \left( A = \{a, b, c, d\}, \quad P = \left\{ \begin{array}{l} (a, b), (b, c), (c, a), \\ (b, c), (c, d), (d, b) \end{array} \right\}, \quad k = 4 \right)$$

Let me show that the mapping from instances of Cyclic Ordering into instances of MaxOrdering defined in (63) is a reduction algorithm, according to (43). Clearly, this mapping is computable in time polynomial in  $|A|$ . Thus, I only need to show that an instance  $(A, T)$  of CyclicOrdering admits a positive answer iff the corresponding instance  $(A, P, k)$  of MaxOrdering admits a positive answer. To start, assume that the given instance  $(A, T)$  of the Cyclic Ordering problem admits a positive answer. Thus,  $T$  is cyclically compatible with a linear order  $\pi$  on  $A$ . In other words, for every triplet  $(a, b, c) \in T$ , there are at least two pairs in  $P$  compatible with  $\pi$ . Hence, there is a multi-set  $S$  of pairs of  $P$  with cardinality at least  $k = 2|T|$  linearly compatible with  $\pi$ ,<sup>16</sup> This conclusion says that the instance of the MaxOrdering problem defined in (63) admits a positive answer too. Vice versa, assume that the instance  $(A, P, k)$  of the MaxOrdering problem in (63) admits a positive answer. Thus, there exists a linear order  $\pi$  on  $A$  compatible with  $2|T|$  pairs in  $P$ . Since the three pairs that come from a given triplet are inconsistent, then each triplet must contribute two pairs to the total of  $2|T|$  compatible pairs. In other words,  $\pi$  is cyclically compatible with all triplets in  $T$ . This conclusion says that the given instance  $(A, T)$  of the Cyclic Ordering problem admits a positive answer.  $\square$

**Lemma 4.** *The MaxOrdering problem (61) is reducible to the PT-Restrictiveness problem (57).*

*Proof.* Given an instance  $(A, P, k)$  of the MaxOrdering problem, construct the corresponding instance  $(\mathbf{A}, \mathcal{F}, K)$  of the PT-Restrictiveness problem as follows. Let  $n = |A|$  and  $\ell = |P|$ . Pick an integer  $d$  as in (65a). Let the threshold  $K$  be defined as in (65b); let the numbers  $N$  and  $M$  of columns and rows of the tableau  $\mathbf{A}$  be as in (65c).

$$(65) \quad \text{a. } d > (\ell - k)n$$

$$\text{b. } K = (\ell - k)(n + d)$$

$$\text{c. } N = \ell + n + d$$

$$M = \ell + nd$$

Let the sets  $\mathcal{F}$  and  $\mathcal{M}$  of faithfulness and markedness constraints be as in (66). There is a faithfulness constraint  $F_{(i,j)}$  for every pair  $(a_i, a_j)$  in the multi-set  $P$  in the given instance of the MaxOrdering problem. Markedness constraints come in two varieties. There are the markedness constraints  $M_1, \dots, M_n$ , one for every element in the set  $A$  given with the instance of the MaxOrdering problem; and then there are  $d$  more markedness constraints  $M'_1, \dots, M'_d$ , that I'll call the *ballast* markedness constraints.

$$(66) \quad \mathcal{F} = \{F_{(i,j)} \mid (a_i, a_j) \in P\}$$

$$\mathcal{M} = \{M_1, \dots, M_n\} \cup \{M'_1, \dots, M'_d\}$$

<sup>15</sup>A similar claim appears (without proof) in Cohen et al. (1999).

<sup>16</sup>Note that, in order for the latter claim to hold, it is crucial that  $P$  be a multi-set, namely that the same pair might be counted twice. In fact,  $T$  might contain two different triplets that share some elements, such as  $(a, b, c)$  and  $(a, b, d)$ .

The comparative tableau  $\mathbf{A}$  is built by assembling one underneath the other various blocks. To start, let  $\overline{\mathbf{A}}$  be the block with  $\ell$  rows and  $N = \ell + n + d$  columns described in (??). It has a row for every pair  $(a_i, a_j)$  in the multi-set  $P$ . This row has all E's but for three entries: the entry corresponding to the faithfulness constraint  $F_{(i,j)}$  corresponding to that pair, which is a w; the entry corresponding to the markedness constraint  $M_i$  corresponding to the first element  $a_i$  in the pair, which is an L; the entry corresponding to the markedness constraint  $M_j$  corresponding to the second element  $a_j$  in the pair, which is a w.

$$(67) \quad (a_i, a_j) \in P \Rightarrow \left[ \begin{array}{ccc|ccc|ccc} \dots & F_{(i,j)} & \dots & \dots & M_i & \dots & M_j & \dots & M'_1 & \dots & M'_d \\ \vdots & \vdots \\ \dots & W & \dots & \dots & L & \dots & W & \dots & E & \dots & E \\ \vdots & \vdots \end{array} \right]$$

I illustrate this construction in (68). Consider the instance of the MaxOrdering problem in (68a): the set  $A$  contains  $n=3$  elements; the multi-set  $P$  contains  $\ell = 3$  pairs; the threshold is set to  $k = 2$ . Pick  $d = (\ell - k)n + 1 = 4$ . The corresponding small comparative tableau  $\overline{\mathbf{A}}$  constructed according to (??) is (68b), where as usual I omit E's for readability.

$$(68) \quad \text{a. } \left( A = \{a, b, c\}, \quad P = \{(a, b), (b, c), (c, a)\} \right)$$

$$\text{b. } \left[ \begin{array}{ccc|ccc|cccc} F_{(a,b)} & F_{(b,c)} & F_{(c,a)} & M_a & M_b & M_c & M'_1 & M'_2 & M'_3 & M'_4 \\ (a,b) & W & & L & W & & & & & \\ (b,c) & & W & & L & W & & & & \\ (c,a) & & & W & & L & & & & \end{array} \right]$$

Next, let  $\mathbf{A}_i$  be the block with  $d$  rows and  $N = \ell + n + d$  columns described in (69), for every  $i = 1, \dots, n$ . All entries corresponding to the faithfulness constraints are equal to E. All entries corresponding to the the markedness constraints  $M_1, \dots, M_n$  are equal to E, but for those in the column corresponding to  $M_i$ , that are instead equal to w. All entries corresponding to the ballast constraints  $M'_1, \dots, M'_d$  are equal to E, but for the diagonal entries that are instead equal to L.

$$(69) \quad \left[ \begin{array}{ccc|ccc|ccc} F_1 & \dots & F_\ell & M_1 & \dots & M_i & \dots & M_n & M'_1 & \dots & M'_d \\ E & \text{---} & E & & & W & & & L & & \\ | & & | & & & | & & & \diagdown & & \\ E & \text{---} & E & & & W & & & & & L \end{array} \right]$$

Finally, the comparative tableau  $\mathbf{A}$  is obtained by ordering the  $n+1$  blocks  $\overline{\mathbf{A}}, \mathbf{A}_1, \dots, \mathbf{A}_n$  just defined one underneath the other, as in (70). The resulting tableau has  $M$  rows and  $N$  columns, with  $M$  and  $N$  as defined in (65c). Let me show that the mapping from instances of the MaxOrdering problem into instances of the PT-Restrictiveness problem defined in (65)-(70) is a reduction algorithm, according to (43). Clearly, this mapping can be computed in polynomial time. In the rest of the proof, I thus focus on showing that the given instance  $(A, P, k)$  of the MaxOrdering problem admits a positive answer iff the corresponding instance  $(\mathbf{A}, \mathcal{F}, K)$  of the PT-Restrictiveness problem admits a positive answer.

$$(70) \quad \left[ \begin{array}{ccc|ccc|ccc} F_1 & \dots & F_\ell & M_1 & \dots & M_n & M'_1 & \dots & M'_d \\ \overline{\mathbf{A}} \\ \hline \mathbf{A}_1 & \left[ \begin{array}{ccc|ccc} E & \text{---} & E & W & & L \\ | & & | & | & & \diagdown \\ E & \text{---} & E & W & & L \end{array} \right] \\ \hline \vdots & & \vdots & & & \vdots & & & \vdots \\ \hline \mathbf{A}_n & \left[ \begin{array}{ccc|ccc} E & \text{---} & E & W & & L \\ | & & | & | & & \diagdown \\ E & \text{---} & E & W & & L \end{array} \right] \end{array} \right]$$

Before I turn to the details, let me present the core intuition. Since the markedness constraints  $M_1, \dots, M_n$  correspond to the elements  $a_1, \dots, a_n$  of  $A$ , a linear order  $\pi$  over  $A$  defines a ranking  $\gg$

of the markedness constraint  $M_1, \dots, M_n$  as in (71), and vice-versa. Thus,  $\pi$  is linearly compatible with a pair  $(a_i, a_j) \in P$  iff the row of the block  $\bar{\mathbf{A}}$  in (??) corresponding to that pair is accounted for by ranking  $M_j$  above  $M_i$ , with no need for the corresponding faithfulness constraint  $F_{(i,j)}$  to do any work. Suppose instead that  $M_j$  is not ranked above  $M_i$ , so that the corresponding faithfulness constraint  $F_{(i,j)}$  needs to be ranked above  $M_i$  in order to protect its L. What consequences does this fact have for the PT-measure  $\mu_{\text{PT}}$  in (31)? Without the ballast constraints  $M'_1, \dots, M'_d$ , not much: all I could conclude is that the faithfulness constraint  $F_{(i,j)}$  has at least the two markedness constraints  $M_i$  and  $M_j$  ranked below it. The ballast markedness constraints  $M'_1, \dots, M'_d$  ensure a more dramatic effect. In fact, the block  $\mathbf{A}_i$  forces each of them to be ranked below  $M_i$ . Thus, if the faithfulness constraint  $F_{(i,j)}$  needs to be ranked above  $M_i$ , then it also needs to be ranked above all the ballast markedness constraints  $M'_1, \dots, M'_d$ . If the number  $d$  of these ballast constraints is large enough, as in (65a), then the corresponding effect on the PT-measure  $\mu_{\text{PT}}$  turns out to be rather dramatic. The rest of the proof formalizes this intuition.

$$(71) \quad M_j \gg M_i \iff \pi(a_j) > \pi(a_i)$$

To start, assume that the given instance  $(A, P, k)$  of the MaxOrdering problem admits a positive answer. Thus, there exists a multi-set  $S$  consisting of  $k$  pairs of  $P$  that is compatible with a linear order  $\pi$  on  $A$ . Consider a ranking  $\gg$  over the constraint set (66) that satisfies the conditions in (72):  $\gg$  assigns the  $k$  faithfulness constraints  $F_{(i,j)}$  that correspond to pairs in  $S$  to the  $k$  bottom strata, in any order;  $\gg$  assigns the  $d$  ballast markedness constraints  $M'_1, \dots, M'_d$  to the next  $d$  strata, in any order;  $\gg$  assigns the  $n$  markedness constraints  $M_1, \dots, M_n$  to the next  $n$  strata, ordered according to  $\pi$  through (71); finally,  $\gg$  assigns the remaining  $\ell - k$  faithfulness constraints  $F_{(i,j)}$  that correspond to pairs in  $P \setminus S$  to the top  $\ell - k$  strata, in any order.

$$(72) \quad \begin{array}{ll} \{F_{(i,j)} \mid (a_i, a_j) \notin S\} & \text{arbitrarily ranked} \\ | & \\ \{M_1, \dots, M_n\} & \text{ranked according to } \pi \text{ through (71)} \\ | & \\ \{M'_1, \dots, M'_d\} & \text{arbitrarily ranked} \\ | & \\ \{F_{(i,j)} \mid (a_i, a_j) \in S\} & \text{arbitrarily ranked} \end{array}$$

This ranking  $\gg$  is OT-consistent with the comparative tableau  $\mathbf{A}$  in (70). In fact, it is OT-consistent with the  $n$  blocks  $\mathbf{A}_1, \dots, \mathbf{A}_n$  in (69), since the markedness constraints  $M_1, \dots, M_n$  are  $\gg$ -ranked above the ballast markedness constraints  $M'_1, \dots, M'_d$ . It is OT-consistent with each row of the block  $\bar{\mathbf{A}}$  in (??) that corresponds to a pair  $(a_i, a_j) \notin S$ , since the corresponding faithfulness constraint  $F_{(i,j)}$  is  $\gg$ -ranked above the corresponding markedness constraints  $M_i$ . Finally, it is OT-consistent with each row of the block  $\bar{\mathbf{A}}$  that corresponds to a pair  $(a_i, a_j) \in S$ , since  $\pi(a_j) > \pi(a_i)$  and thus  $M_j \gg M_i$  by (71). The PT-measure  $\mu_{\text{PT}}(\gg)$  of the ranking  $\gg$  is (73): in fact, the faithfulness constraints  $F_{(i,j)}$  corresponding to pairs  $(a_i, a_j) \in S$  have no markedness constraints  $\gg$ -ranked below them; and each one of the  $\ell - k$  faithfulness constraints  $F_{(i,j)}$  corresponding to pairs  $(a_i, a_j) \notin S$  has all the  $n + d$  markedness constraints  $\gg$ -ranked below it. In conclusion, the instance  $(\mathbf{A}, \mathcal{F}, K)$  of the PT-Restrictiveness problem constructed in (65)-(70) admits a positive answer.

$$(73) \quad \mu_{\text{PT}}(\gg) = (\ell - k)(n + d) = K$$

Vice versa, assume that the instance  $(\mathbf{A}, \mathcal{F}, K)$  of the PT-Restrictiveness problem constructed in (65)-(70) admits a positive answer. Thus, there exists a ranking  $\gg$  over the constraint set (66) OT-consistent with the tableau  $\mathbf{A}$  in (70) which furthermore has PT-strictness measure  $\mu_{\text{PT}}(\gg)$  at most  $K$ . Consider the multi-set  $S \subseteq P$  defined in (74). Clearly,  $S$  is compatible with the linear order  $\pi$  univocally defined on  $A = \{a_1, \dots, a_n\}$  through (71).

$$(74) \quad S = \left\{ (a_i, a_j) \in P \mid M_j \gg M_i \right\}$$

To prove that the given instance  $(A, P, k)$  of the MaxOrdering problem has a positive answer, I thus only need to show that  $|S| \geq k$ . Assume by contradiction that  $|S| < k$ . I can then compute as in (75). In step (75a), I have used the definition (65b) of the threshold  $K$ . In step (75b), I

have used the hypothesis that the ranking  $\gg$  is a solution of the instance  $(\mathbf{A}, \mathcal{F}, K)$  of the PT-Restrictiveness problem and thus its PT- measure  $\mu_{\text{PT}}(\gg)$  does not exceed  $K$ . By (31),  $\mu_{\text{PT}}(\gg)$  is the total number of pairs of a faithfulness constraint and a markedness constraint such that the former is  $\gg$ -ranked above the latter. In step (75c), I have thus lower bounded  $\mu_{\text{PT}}(\gg)$  by only considering those faithfulness constraints  $F_{(i,j)}$  corresponding to pairs  $(a_i, a_j)$  that do not belong to  $S$ . For each such constraint  $F_{(i,j)}$ , we have  $M_i \gg M_j$ , by the definition (74) of  $S$ . Thus,  $F_{(i,j)}$  needs to be  $\gg$ -ranked above  $M_i$  in order to ensure OT-consistency with the corresponding row of the block  $\overline{\mathbf{A}}$  in (??). Since in turn  $M_i$  needs to be  $\gg$ -ranked above the  $d$  ballast constraints  $M'_1, \dots, M'_d$  in order to ensure OT-consistency with the block  $\mathbf{A}_i$  in (69), then  $F_{(i,j)}$  needs to be  $\gg$ -ranked above those  $d$  ballast markedness constraints too. In conclusion, each faithfulness constraint  $F_{(i,j)}$  corresponding to a pair  $(a_i, a_j)$  that does not belong to  $S$  needs to be  $\gg$ -ranked at least above  $d$  markedness constraints. Since there are  $\ell - |S|$  such faithfulness constraints  $F_{(i,j)}$  corresponding to pairs  $(a_i, a_j)$  that do not belong to  $S$ , then we get the inequality in (75d). In step (75e), I have used the absurd hypothesis that  $|S| < k$  or equivalently that  $|S| \leq k - 1$ . The chain of inequalities in (75) entails that  $d \leq (\ell - k)n$ , which contradicts the choice (65a) of the number  $d$  of ballast constraints.

$$\begin{aligned}
(75) \quad & (\ell - k)d + (\ell - k)n \\
& \stackrel{(a)}{\geq} K \\
& \stackrel{(b)}{\geq} \mu_{\text{PT}}(\gg) \stackrel{(31)}{=} |\{(F_{(i,j)}, M) \mid F_{(i,j)} \gg M\}| \\
& \stackrel{(c)}{\geq} |\{(F_{(i,j)}, M) \mid F_{(i,j)} \gg M, (a_i, a_j) \notin S\}| \\
& \stackrel{(d)}{\geq} (\ell - |S|)d \\
& \stackrel{(e)}{\geq} (\ell - (k - 1))d \\
& = (\ell - k)d + d
\end{aligned}$$

The preceding considerations show that an instance  $(A, P, k)$  of the MaxOrdering problem (61) admits a positive answer iff the corresponding instance  $(\mathbf{A}, \mathcal{F}, K)$  of the PT-Restrictiveness problem (57) defined in (65)-(70) admits a positive answer. I conclude that the MaxOrdering problem can be reduced to the PT-Restrictiveness problem.  $\square$

**Lemma 5.** *The PT-Restrictiveness problem (57) is reducible to the Restrictiveness problem (54).*

*Proof.* Given an instance  $(\mathbf{A}, \mathcal{F}, k)$  of the PT-Restrictiveness problem (57), construct the corresponding instance  $((\mathcal{X}, \mathcal{Y}, \text{Gen}, C), D, K)$  of the Restrictiveness problem (54) as follows. Let  $m$  and  $n$  be the number of rows and of columns of the comparative tableau  $\mathbf{A}$ ; let  $\ell$  be the cardinality of the set  $\mathcal{F}$  of faithfulness constraints; finally, let  $d = \ell(n - \ell)$ . The threshold  $K$  is defined as in (76a). The sets  $\mathcal{X}$  and  $\mathcal{Y}$  of underlying and surface forms are defined as in (76b), so that underlying and surface forms are sorted into three classes  $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$  and  $\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3$ . The generating function  $\text{Gen}$  is defined as in (76c). The data set  $D$  is defined as in (76d).

$$\begin{aligned}
(76) \quad & \text{a. } K = m + k + d \\
& \text{b. } \mathcal{X} = \{x_1, \dots, x_m\} \cup \{x'_1, \dots, x'_d\} \cup \{x''_1, \dots, x''_d\} \\
& \quad \begin{array}{ccc}
| & | & | \\
\mathcal{X}_1 & \mathcal{X}_2 & \mathcal{X}_3
\end{array} \\
& \mathcal{Y} = \left\{ \begin{array}{l} y_1, \dots, y_m \\ z_1, \dots, z_m \end{array} \right\} \cup \left\{ \begin{array}{l} u_1, \dots, u_d \\ w_1, \dots, w_d \end{array} \right\} \cup \left\{ \begin{array}{l} u_1, \dots, u_d \\ v_1, \dots, v_d \end{array} \right\} \\
& \quad \begin{array}{ccc}
| & | & | \\
\mathcal{Y}_1 & \mathcal{Y}_2 & \mathcal{Y}_3
\end{array} \\
& \text{c. } \text{Gen}(x_i) = \{y_i, z_i\} \subseteq \mathcal{Y}_1 \quad \text{for } x_i \in \mathcal{X}_1 \\
& \quad \text{Gen}(x'_i) = \{u_i, w_i\} \subseteq \mathcal{Y}_2 \quad \text{for } x'_i \in \mathcal{X}_2 \\
& \quad \text{Gen}(x''_i) = \{u_i, v_i\} \subseteq \mathcal{Y}_3 \quad \text{for } x''_i \in \mathcal{X}_3 \\
& \text{d. } D = \{(x_1, y_1), \dots, (x_m, y_m)\} \subseteq \mathcal{X} \times \mathcal{Y}
\end{aligned}$$

Let the constraint set  $C$  contain a total of  $n$  constraints  $C_1, \dots, C_n$ ; let  $C_h$  be a faithfulness constraint iff  $h \in \mathcal{F}$ , and a markedness constraint otherwise. Since,  $\text{Gen}(\mathcal{X}_i) \subseteq \mathcal{Y}_i$  for  $i = 1, 2, 3$ ,

constraints need only be defined on  $\mathcal{X}_i \times \mathcal{Y}_j$  with  $i=j$ . The set  $\mathcal{X}_1$  contains  $m$  underlying forms  $x_1, \dots, x_m$ , one for every row of the given comparative tableau  $\mathbf{A}$ . Each of these underlying forms  $x_i$  comes with the two candidates  $y_i$  and  $z_i$  in  $\mathcal{Y}_1$ . The data set  $D$  in (76d) is a subset of  $\mathcal{X}_1 \times \mathcal{Y}_1$ . Define the constraints  $C_1, \dots, C_n$  over  $\mathcal{X}_1 \times \mathcal{Y}_1$  as in (77a). This definition ensures that  $\mathbf{A}$  is the comparative tableau corresponding to  $D$ , so that condition (77b) holds for any ranking.

- (77) a.  $C_h(x_i, y_i) < C_h(x_i, z_i) \iff$  the  $k$ th entry in the  $i$ th row of  $\mathbf{A}$  is a w  
 $C_h(x_i, y_i) = C_h(x_i, z_i) \iff$  the  $k$ th entry in the  $i$ th row of  $\mathbf{A}$  is a e  
 $C_h(x_i, y_i) > C_h(x_i, z_i) \iff$  the  $k$ th entry in the  $i$ th row of  $\mathbf{A}$  is a l  
 b.  $\gg$  is OT-consistent with the tableau  $\mathbf{A}$  iff  $\gg$  is OT-consistent with the data set  $D$ .

Next, define the constraints  $C_1, \dots, C_n$  over  $\mathcal{X}_2 \times \mathcal{Y}_2$  as in (78a). This definition ensures that the forms  $u_1, \dots, u_d$  are *unmarked*, and thus belong to the language corresponding to any ranking  $\gg$ , as stated in (78b).

- (78) a.  $C_h(x'_i, u_i) \leq C_h(x'_i, w_i)$  for every constraint  $C_h$   
 b.  $\{u_1, \dots, u_d\} \subseteq \mathcal{L}(\gg)$

The set  $\mathcal{X}_3$  contains a total of  $d = \ell(n - \ell)$  underlying forms  $x''_1, \dots, x''_2$ , one for every pair of a faithfulness constraint and a markedness constraint. Pair up (in some arbitrary but fixed way) each of these underlying forms with a unique pair of a faithfulness constraint and a markedness constraint. Thus, I can speak of “the” markedness constraint and “the” faithfulness constraint “corresponding” to a given underlying form  $x''_i \in \mathcal{X}_3$ . Each of these underlying forms  $x''_i$  comes with two candidates  $u_i$  and  $v_i$  in  $\mathcal{Y}_3$ . Define the constraints  $C_1, \dots, C_n$  over  $\mathcal{X}_3 \times \mathcal{Y}_3$  as in (79a). This definition ensures that the grammar  $\text{OT}\gg$  corresponding to an arbitrary ranking  $\gg$  maps  $x''_i$  to  $v_i$  rather than to  $u_i$  iff the faithfulness constraint corresponding to the underlying form  $x''_i$  is  $\gg$ -ranked above the markedness constraint corresponding to  $x''_i$ . Since  $\mu_{\text{PT}}(\gg)$  is defined in (31) as the total number of pairs of a faithfulness and a markedness constraint such that the former is ranked above the latter, then condition (79b) holds for any ranking: the PT-measure of a ranking coincides with the number of forms  $v_1, \dots, v_d$  that belong to the corresponding language. The idea here is that rankings that correspond to small languages are those that map as many underlying forms  $x''_i \in \mathcal{X}_3$  to the candidate  $u_i$  rather than to the candidate  $v_i$ , as the candidate  $u_i$  belongs to the language no matter what by (78b).

- (79) a.  $C_h(x''_i, v_i) < C_h(x''_i, u_i)$  if  $C_h$  is the faithfulness constraint corresponding to  $x''_i$   
 $C_h(x''_i, v_i) > C_h(x''_i, u_i)$  if  $C_h$  is the markedness constraint corresponding to  $x''_i$   
 $C_h(x''_i, v_i) = C_h(x''_i, u_i)$  otherwise  
 b.  $\mu_{\text{PT}}(\gg) = |\mathcal{L}(\gg) \cap \{v_1, \dots, v_d\}|$

Let me show that the mapping from instances of the PT-Restrictiveness problem into instances of the Restrictiveness problem defined in (76)-(79) is a reduction algorithm, according to (43). Clearly, this mapping can be computed in polynomial time. In the rest of the proof, I thus only need to show that a given instance  $(\mathbf{A}, \mathcal{F}, k)$  of the PT-Restrictiveness problem admits a positive answer iff the corresponding instance  $((\mathcal{X}, \mathcal{Y}, \text{Gen}, \mathcal{C}), D, K)$  of the Restrictiveness problem admits a positive answer. To start, assume that the original instance  $(\mathbf{A}, \mathcal{F}, k)$  of the PT-Restrictiveness problem admits a positive answer. Thus, there exists a ranking  $\gg$  OT-consistent with the comparative tableau  $\mathbf{A}$  whose PT-measure  $\mu_{\text{PT}}(\gg)$  is at most  $k$ . Since  $\gg$  is OT-consistent with  $\mathbf{A}$ , then  $\gg$  is OT-consistent with  $D$ , by (77b). Furthermore, the language  $\mathcal{L}(\gg)$  corresponding to the ranking  $\gg$  contains at most  $K = m + k + d$  surface forms, namely: the  $m$  surface forms  $y_1, \dots, y_m \in \mathcal{Y}_1$ , because  $\gg$  is OT-consistent with  $D$ ; the  $d$  surface forms  $u_1, \dots, u_d$ , by (78b); and at most  $k$  of the surface forms  $v_1, \dots, v_d$ , by (78b) and the hypothesis that  $\mu_{\text{PT}}(\gg)$  is at most  $k$ . In conclusion,  $\gg$  is a solution of the instance  $((\mathcal{X}, \mathcal{Y}, \text{Gen}, \mathcal{C}), D, K)$  of the Restrictiveness problem constructed in (76)-(79). The same reasoning shows that the vice versa holds too.  $\square$

## REFERENCES

- Angluin, Dana. 1980. “Inductive Inference of formal languages from positive data”. *Information Control* 45:117–135.

- Arora, Sanjeev, Alan Frieze, and Haim Kaplan. 2002. “A New Rounding Procedure for the Assignment Problem with Applications to Dense Graph Arrangement Problems”. *Mathematical Programming* 92.1:1–36.
- Beckman, Jill N. 1997. “Positional faithfulness, positional neutralization and Shona vowel harmony”. *Phonology* 14:1–46.
- Bertsimas, Dimitris, and Robert Weismantel. 2005. *Optimization over Integers*. Belmont, Massachusetts: Dynamic Ideas.
- Berwick, Robert. 1985. *The acquisition of syntactic knowledge*. Cambridge, MA: MIT Press.
- Cohen, W., William, Robert E. Schapire, and Yoram Singer. 1999. Learning to order things. *Journal of Artificial Intelligence Research* 10:243–270.
- Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. 1990. *Introduction to Algorithms*. Cambridge, MA: MIT Press.
- Eisner, Jason. 1997. “Efficient Generation in Primitive Optimality Theory”. In *Proceedings of the 35th Annual ACL and 8th European ACL*, 313–320. Madrid, Spain.
- Eisner, Jason. 2000. “Easy and Hard Constraint Ranking in Optimality Theory”. In *Finite-State Phonology: Proceedings of the Fifth Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, ed. J. Eisner, L. Karttunen, and A. Thériault, 22–33. Luxembourg.
- Galil, Zvi, and Nimrod Megiddo. 1977. “Cyclic Ordering is NP-complete”. *Theoretical Computer Science* 5:179–182.
- 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.
- Hayes, Bruce. 2004. “Phonological Acquisition in Optimality Theory: The Early Stages”. In *Constraints in Phonological Acquisition*, ed. R. Kager, J. Pater, and W. Zonneveld, 158–203. Cambridge University Press.
- Heinz, Jeffrey, Gregory M. Kobele, and Jason Riggle. 2009. “Evaluating the Complexity of Optimality Theory”. *Linguistic Inquiry* 40:277–288.
- Heinz, Jeffrey, and Jason Riggle. to appear. “Learnability”. In *Blackwell Companion to Phonology*, ed. Marc van Oostendorp, Colin Ewen, Beth Hume, and Keren Rice. Wiley-Blackwell.
- Idsardi, William. 2006. “A simple proof that Optimality Theory is computationally intractable”. *Linguistic Inquiry* 37.2:271–275.
- Jun, Jongho. 2004. “Place assimilation”. In *Phonetically Based Phonology*, ed. B. Hayes, R. Kirchner, and D. Steriade, 58–86. Cambridge University Press.
- Kaun, Abigail Rhoades. 1995. The typology of Rounding Harmony: An Optimality Theoretic Approach. Doctoral Dissertation, UCLA.
- Legendre, Géraldine, Yoshiro Miyata, and Paul Smolensky. 1990a. “Harmonic Grammar: A formal multi-level connectionist theory of linguistic well-formedness: An application”. In *Proceedings of the twelfth annual conference of the Cognitive Science Society*, 884–891. Cambridge, MA: Lawrence Erlbaum.
- Legendre, Géraldine, Yoshiro Miyata, and Paul Smolensky. 1990b. “Harmonic Grammar: A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations”. In *Proceedings of the twelfth annual conference of the Cognitive Science Society*, 388–395. Cambridge, MA: Lawrence Erlbaum.
- Lombardi, Linda. 1999. “Positional faithfulness and voicing assimilation in Optimality Theory”. *Natural Language and Linguistic Theory* 17:267–302.
- Magri, Giorgio. 2007. “Quadratic Phonotactics”. In *Proceedings of NELS38*.
- Magri, Giorgio. 2008. “An integer programming formulation of the problem of the acquisition of phonotactics in Optimality Theory”. Manuscript, MIT.
- Magri, Giorgio. 2010a. “A computational investigation of OT online models of the early stage of the acquisition of phonotactics. Part 2: correctness”. Manuscript, IJN, ENS.
- Magri, Giorgio. 2010b. Complexity of the acquisition of phonotactics in optimality theory. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, ed. Jeffrey Heinz, Lynne Cahill, and Richard Wicentowski, 19–27. Uppsala, Sweden: Association for Computational Linguistics.
- Manzini, M. Rita, and Ken Wexler. 1987. “Parameters, Binding Theory, and Learnability”. *Linguistic Inquiry* 18.3:413–444.
- Prince, Alan. 2002. “Entailed Ranking Arguments”. ROA 500.

- Prince, Alan, and Paul Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell.
- Prince, Alan, and Bruce Tesar. 2004. "Learning Phonotactic Distributions". In *Constraints in Phonological Acquisition*, ed. R. Kager, J. Pater, and W. Zonneveld, 245–291. Cambridge University Press.
- Steriade, Donca. 1999. "Directional asymmetries in place assimilation. A perceptual account". In *The role of speech perception in phonology*, ed. Hume and Johnson, 219–247. Academic Press.
- Steriade, Donca. 2001. "The phonology of perceptibility: The P-map and its consequences for constraint organization". Ms., UCLA.
- Tesar, Bruce. 1995. "Computational Optimality Theory". Doctoral Dissertation, University of Colorado, Boulder. ROA 90.
- Tesar, Bruce. 2008. "Output-Driven Maps". Ms., Rutgers University; ROA-956.
- Tesar, Bruce, and Paul Smolensky. 1998. "Learnability in Optimality Theory". *Linguistic Inquiry* 29:229–268.
- Tesar, Bruce, and Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge, MA: The MIT Press.
- Wareham, Harold Todd. 1998. Systematic Parameterized Complexity Analysis in Computational Phonology. Doctoral Dissertation, University of Victoria, Dept. of Computer Science.