On-line learning of underlying forms

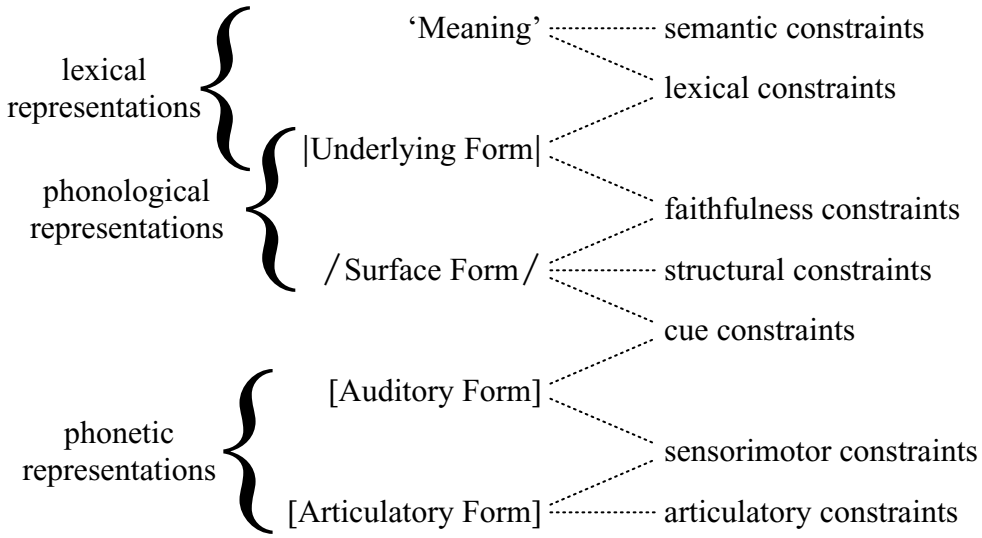
Abstract

This paper shows how underlying forms are learned by an OT on-line learning algorithm. The proposed algorithm is stupid: it processes one form at a time under one grammar at a time. Stupid is good and effective: no extra learning mechanisms are required than the ones already involved in a general grammar learning model of OT (e.g. Boersma 1997). Interpretation of incoming forms and constraint reranking as a reaction to error detection is enough. Surface and underlying forms are connected via faithfulness constraints, and underlying forms are connected to meaning via lexical constraints. It is shown that the learner can acquire grammar and underlying forms concurrently, and creates an economical lexicon. This is exemplified with the learning of lexical stress. Lexical stress (in opposition to grammatically assigned stress) is information that is not predictable by the grammar (i.e. the constraint ranking of a language), and is therefore stored in the lexicon as some sort of underlying representation.

1 Introduction

The proposed grammar model is based on Boersma (2005), but extended with an additional representational level 'Meaning', as also discussed in Boersma (2006a, b). In this model, the different levels of representations are connected through different groups of constraints:

(1) The grammar model:



This paper deals with the top three levels of representation *meaning*, *underlying form* and *surface form*. The connection between surface and underlying forms is expressed by the well-known family of faithfulness constraints. The connection between form and meaning is expressed by a new family of lexical constraints. The term 'meaning' as used here conflates both semantic and syntactical information.

The next section outlines lexical stress in Modern Greek, while section 3 outlines the learning model. The model is put to the test in computer simulations, shown in section 4. The results are given in section 5. Section 6 discusses some alternatives, and section 7 provides a general discussion.

2 Lexical stress

The learning of underlying forms is illustrated here with a simplified version of Modern Greek stress. In Modern Greek, stress is mainly assigned through specifications in the lexicon. Roots as well as suffixes can contrast in stress. Roots can be specified for being unstressed, stressed or post-stressing. Suffixes can be specified for being unstressed, stressed, or pre-stressing. In the case that a word is built out of morphemes without any specification for stress, a phonological default applies and stress is assigned on the antepenultimate syllable of the word. We will focus on the contrast between underlyingly stressed/unstressed morphemes. For instance, a word like *ɣóndola* 'gondola-NOM.SG' remains stressed on the root when inflected with the genitive plural suffix *-on*: *ɣóndolon*. But a word like *θálasa* 'sea-NOM.SG' shifts stress to the suffix when inflected: *θalasón*. A word like *ɣóndola* is traditionally analyzed as being underlyingly stressed on the root |ɣóndol-|, and a word like *θálasa* is analyzed as being underlyingly unstressed |θalas-|. The genitive plural suffix *-on* is underlyingly stressed, as becomes apparent when attached to an unstressed root like in the case of *θalasón*: only then it can surface as stressed. The nominative singular suffix *-a* is unstressed, as becomes apparent when combined with an unstressed root *θalas-*: then the phonological default is assigned. A learner of Modern Greek would have to find out that the root *ɣondol-* is underlyingly stressed, represented as |ɣóndol-|, while the root *θalas-* is underlyingly unstressed, represented as |θalas-|. Likewise with the suffixes: *-on* should be underlyingly represented as stressed |-ón|, and *-a* should be underlyingly represented as unstressed |-a|. We will ignore foot structure for this simplified version of Modern Greek stress, and represent surface forms (SF) as in (2a). Underlying forms (UF) are represented as in (2b), and meaning as in (2c).

(2) Modern Greek:

a. surface forms	b. underlying forms	c. meaning
/ɣóndola/	|ɣóndol+a|	'gondola-NOM.SG'
/ɣóndolon/	|ɣóndol+ón|	'gondola-GEN.PL'
/θálasa/	|θalas+a|	'sea-NOM.SG'
/θalasón/	|θalas+ón|	'sea-GEN.PL'

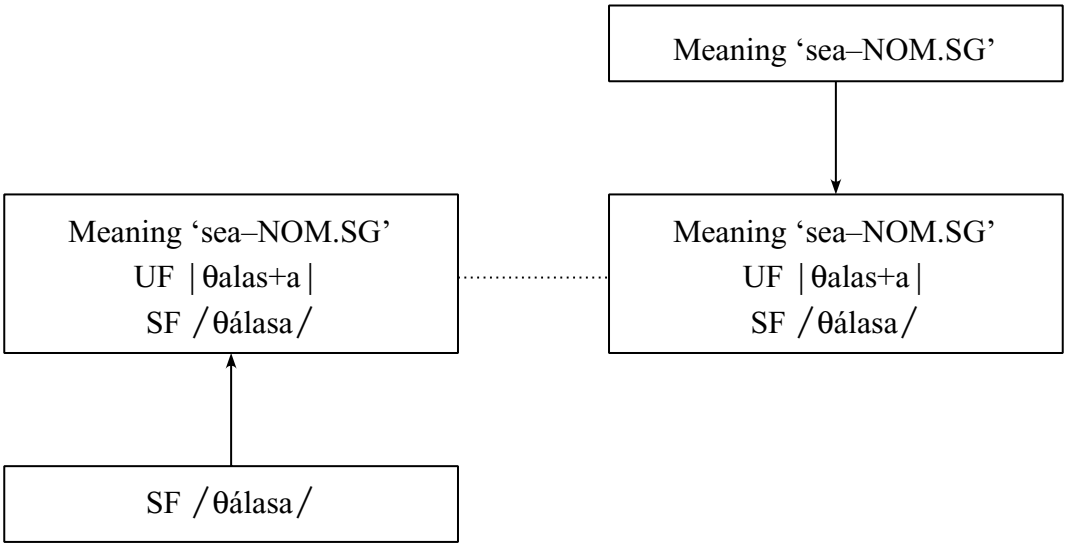
Ideally, the learner should end up with four underlying forms: |ɣóndol-|, |θalas-|, |-ón|, |-a|.

3 The learning model

In the proposed model the learner has to both learn the ranking of constraints and the underlying forms for a language. The SF is what is observable to the learner (as in (2a)). Given is a set of constraints, connecting SFs to UFs (UFs as in (2b) above), and UFs to

meaning (meaning as in (2c) above). The mapping between SF and UF is regulated by well-known faithfulness constraints (4), and the mapping between UF and meaning is regulated by lexical constraints (5). The learning process is *error-driven* (Wexler & Culicover 1980:127), in the sense that a learner compares the form she recognizes with what she would produce for that word. In recognition, the SF serves as input to the OT evaluation, and the candidates are triplets of meaning, UFs, and SFs. In virtual production, i.e. the computation of the produced form, meaning serves as input to the evaluation, and candidates are the same triplets of UFs and SFs as in recognition.[1] If the produced meaning/UF/SF pair is the same as the one in recognition, nothing in the grammar is changed. If there is a mismatch (i.e., error detection) the grammar (i.e., the ranking) is changed.

(3) The processing model:



3.1 The constraints

We propose two kinds of constraints here that are active in the learning of underlying forms. On one hand there are faithfulness constraints establishing a correspondence between SF and UF, in this case faithfulness constraints on stress:

(4) Faithfulness constraints:
MAX(R): A stressed root in the underlying form is also stressed in the surface form.
DEP(R): A stressed root in the surface form is also stressed in the underlying form.
MAX(A): A stressed affix in the underlying form is also stressed in the surface form.
DEP(A): A stressed affix in the surface form is also stressed in the underlying form.

[1] This does not mean that the UF is actually produced; it simply means that meaning, underlying form and surface form are evaluated in parallel.

The other group of constraints are lexical constraints and are formulated as 'don't connect the meaning 'xy' to the form |XY| specified/unspecified for stress'. They establish the link between UF and meaning.

(5) Lexical constraints:
*|ɣondol-| 'gondola': Don't connect the meaning 'gondola' to an unstressed root |ɣondol-|.
*|ɣóndol-| 'gondola': Don't connect the meaning 'gondola' to a stressed root |ɣóndol-|.
*|θalas-| 'sea': Don't connect the meaning 'sea' to an unstressed root |θalas-|.
*|θálas-| 'sea': Don't connect the meaning 'sea' to a stressed root |θálas -|.
*|-a| 'NOM.SG': Don't connect the meaning 'NOM.SG' to an unstressed suffix |-a|.
*|-á| 'NOM.SG': Don't connect the meaning 'NOM.SG' to a stressed suffix |-á|.
*|-on| 'GEN.PL': Don't connect the meaning 'GEN.PL' to an unstressed suffix |-on|.
*|-ón| 'GEN.PL': Don't connect the meaning 'GEN.PL' to a stressed suffix |-ón|.

Note that the constraint set should in principle also contain constraints like *|ɣondol-| 'sea' ('don't connect a meaning 'sea' to the root |ɣondol-|'). We saved some ink and computation time by not including these constraints, since they would always end up top-ranked in the computer simulations presented in section 4.

3.2 The recognition mapping

The learning process proceeds as follows: the learner interpretes an incoming SF as an UF in the fashion of *robust interpretive parsing* (Tesar 1995, 1997, Tesar & Smolensky 1996, 2000) or *perception* (Boersma 1997 et seq.) of overt forms. This means that the SF becomes the input to an OT evaluation, with triplets of meaning/UF/SF as candidates (provided by GEN). A ranking of faithfulness constraints in interaction with lexical constraints determines the optimal meaning/UF/SF triplet. In (6), the evaluation of an incoming SF is shown. The candidates consist of meaning/UF/SF triplet that only differ in the UF. The UFs are split up into roots and affixes, and since there is a two-way contrast in roots and affixes (stressed/unstressed), there are four possible candidates.[2] Imagine that at some point in learning, the constraint ranking is as in (6). Any unfaithful candidate is ruled out, and the winner is the one with an underlyingly unstressed root and an underlyingly stressed affix.

(6) Recognition of *θalasón* with a given ranking:

SF / θalasón /	MAX (R)	MAX (A)	DEP (R)	DEP (A)	*|θalas-| 'sea'	*|θálas-| 'sea'	*|-on| 'Gen.Pl'	*|-ón| 'Gen.Pl'
'sea-Gen.Pl' | θalas+on | / θalasón /				*!	*		*	
'sea-Gen.Pl' | θálas+on | / θalasón /	*!			*		*	*	
☞ 'sea-Gen.Pl' | θalas+ón | / θalasón /					*			*
'sea-Gen.Pl' | θálas+ón | / θalasón /	*!					*		*

[2] Here we assume that only the initial syllable of the root can be specified for stress. In real-life Greek, however, any of the syllables in disyllabic roots could be specified for stress. We limit ourselves to only one position for the moment.

3.3 Virtual production

Starting out with the meaning, the learner can now compute her production. The meaning becomes the input to the production evaluation. Again, the learner can choose between meaning/UF/SF triplets by applying her grammar (the same constraint ranking as in the recognition step). If the optimal meaning/UF/SF triplet is equal to the one in recognition, no reranking takes place. If there is a mismatch (i.e., error detection), the learner will adjust her constraint ranking (by applying e.g. GLA; Boersma 1997). This is shown in (7): the optimal candidate in production (indexed by the pointing hand) is different from the optimal one in recognition (indexed by an ear; this is the winning candidate of tableau (6)). In production, the candidate with an underlyingly stressed root wins. This brings about constraint reranking.

(7) Production of 'sea-GEN.PL'

'sea-Gen.Pl'	MAX (R)	MAX (A)	DEP (R)	DEP (A)	*|θalas-| 'sea'	*|θálas-| 'sea'	*|-on| 'Gen.Pl'	*|-ón| 'Gen.Pl'
'sea-Gen.Pl' |θalas+on| /θalasón/				*!	*		*	
'sea-Gen.Pl' |θalas+on| /θálason/			*!		*		*	
'sea-Gen.Pl' |θálas+on| /θalasón/	*!			*		*	*	
☞ 'sea-Gen.Pl' |θálas+on| /θálason/						←*	←*	
👂 'sea-Gen.Pl' |θalas+ón| /θalasón/					*!→			*→
'sea-Gen.Pl' |θalas+ón| /θálason/		*!	*		*			*
'sea-Gen.Pl' |θálas+ón| /θalasón/	*!					*		*
'sea-Gen.Pl' |θálas+ón| /θálason/		*!				*		*

There are more candidates in the production evaluation than in the recognition evaluation because in production both the UFs and the SFs can vary. It is important to note that the optimal meaning/UF/SF triplet in the production step has to be the same as the optimal meaning/UF/SF triplet in recognition (i.e., the UF that is chosen in production has to have the same stress specifications as the recognized UF, *and* the SF chosen in production has to have the same stress as the recognized SF). Any deviation elicitates error detection and subsequent constraint reranking.

By encountering a long sequence of various different surface forms the learner's grammar eventually arrives at a ranking that gives her the correct underlying forms. The process described here is, for the moment, limited to the recognition and production steps. This means the learner already knows that stress is lexical. Eventually, the whole acquisition process has to be modelled so as to account for the fact that learners have to acquire whether stress is assigned lexically or by structural principles.

4 Support from computer simulations

We created 10 virtual learners in the Praat program (Boersma & Weenink 1992-2006). All of them had access to the same training data set consisting of four different surface forms (listed in (8)), but encountered the data in a different order (from the training set, a total of 1 000 000 forms were randomly drawn).

(8) The training data:

/ɣóndola/ /ɣóndolon/ /θálasa/ /θalasón/

The learners are furthermore equipped with the constraint set including faithfulness and lexical constraints, and the meaning of the words. GEN provides the learners with the eight possible underlying forms in (9):

(9) Possible underlying forms:

|ɣondol-|	|θalas-|	|-a|	|-on|
|ɣóndol-|	|θálas-|	|-á|	|-ón|

The Gradual Learning Algorithm (GLA; Boersma 1997) was chosen as the reranking strategy (constraints are reranked as in (7)), with weighted uncancelled.[3]

5 Results

All 10 learners chose for underlying representations as displayed in (2b). The ranking of one example learner is shown in (10) (the constraints at the top are ranked higher than the constraints at the bottom):

(10) Ranking learner No. 1:

*|ɣondol-| 'gondola'
*|-á| 'NOM.SG'
MAX(R)
DEP(A)
*|-on| 'GEN.PL'
*|θálas-| 'sea'
*|-a| 'NOM.SG'
*|ɣóndol-| 'gondola'
DEP(R)
MAX(A)
*|-ón| 'GEN.PL'
*|θalas-| 'sea'

All of the learners created grammars with the crucial rankings displayed in (11). Note that it is not important whether a constraint like *|ɣondol-| 'gondola' is ranked above a constraint *|θálas-| 'sea', but only whether they are ranked with another constraint referring to the same meaning.

[3] 'Weighted uncancelled' means that the ranking is lowered for all constraints that are violated more in the recognized form than in the learner's production, and the ranking is raised for all the constraints that are violated more in the learner's production than in the recognized form, and the size of the learning step is divided by the number of moving constraints. This makes sure that the average ranking of all the constraints is constant.

(11) Crucial rankings: [4]

*|ɣondol-| 'gondola' >> *|ɣóndol-| 'gondola'
*|θálas-| 'sea' >> *|θalas-| 'sea'
*|-á| 'NOM.SG' >> *|-a| 'NOM.SG'
*|-on| 'GEN.PL' >> *|-ón| 'GEN.PL'
MAX(R) >> MAX(A)
DEP(A) >> DEP(R)

As a result, the learners choose the four UFs in (12) for their lexicon. This comes about with the ranking: having *|ɣondol-| 'gondola' ranked above *|ɣóndol-| 'gondola' will exclude any candidate from the lexicon that contains an UF |ɣondol-| in connection to the meaning 'gondola'. Likewise, having *|θálas-| 'sea' ranked above *|θalas-| 'sea' will exclude any candidate form the lexicon that contains an UF |θálas-| in connection to the meaning 'sea'.

(12) The resulting lexicon:

|ɣóndol-|	|θalas-|	|-a|	|-ón|

6. Alternatives

In the following, we will briefly discuss some alternatives to the on-line learning approach of underlying forms. In section 6.1 we provide a comparison to Error Driven Constraint Demotion (EDCD; Tesar 1995) instead of the GLA. Section 6.2 outlines Lexicon Optimization as a learning mechanisms fo underlying forms. Section 6.3 explores the off-line learning approach *inconsistency detection* and *surgery* by Tesar et al. (2003). Section 6.4 outlines probabilistic unsupervised learning of underlying forms (Jarosz 2006) that makes use of the Expectation Maximization Algorithm (Dempster et al. 1977).

6.1 Comparison to EDCD

As a comparison, we ran a simulation with 10 virtual learners with EDCD (Tesar 1995) as the reranking strategy. It appeared that the 10 EDCD learners arrived at a ranking that rendered the correct SFs in the production step. However, they decided to create lexical allomorphs: instead of choosing just one morpheme for each meaning, they sometimes chose two, in cases where the surface forms yielded alternation. The root |θalas-| occurred as underlyingly unstressed when combined with the affix |-on|, and as underlyingly stressed when combined with the affix |-a|. The affix |-on| occurred as underlyingly stressed when combined with the root |θalas-|, and as underlyingly unstressed when combined with |ɣóndol-|:

(13) The resulting lexicon of EDCD learners:

|ɣóndol-|	|θálas-|	|-a|	|-on|
	|θalas-|		|-ón|

[4] The crucial rankings displayed here only look like parameter settings: if we took constraints into account that militated against underlying stress on e.g. the second syllable of a root (like we should), then there would be three constraints connected to one meaning, instead of two.

This was due to the fact that the EDCD learners were able to establish a ranking between the lexical constraints, but failed to rank the faithfulness constraints:

(14) The ranking of an example EDCD learner:
{ MAX(R), MAX(A), DEP(R), DEP(A), *|-á| 'NOM.SG', *|ɣondol-| 'gondola' }
>>
{ *|-a| 'NOM.SG', *|-on| 'GEN.PL', *|θálas-| 'sea', *|ɣóndol-| 'gondola' }
>>
{ *|θalas-| 'sea', *|-ón| 'GEN.PL' }

This is a possible solution; however, it is not the most restrictive grammar that can be found. It means that in case of alternation, SFs are always faithfully mapped onto UFs, resulting in an overflow of lexical allomorphs.

6.2 Comparison to Lexicon Optimization

In Lexicon Optimization, as proposed in Prince & Smolensky (1993) and further developed in Itô et al. (1995), the underlying form of a word is determined by evaluating different possible underlying forms with respect to a surface form which is optimal in the ranking of the language. The optimal surface form is determined by the ranking of structural constraints, and the appropriate underlying form for this surface form is determined by faithfulness: the most faithful underlying-surface pair is the most harmonic one, and chosen as the optimal pair. Tesar & Smolensky (1996, 2000) extend Lexicon Optimization by combining the evaluation of different input-output pairs for a given word with input-output pairs of different words, i.e. including paradigmatic comparison.

Lexicon Optimization in its present form is problematic for a learnability approach of underlying forms, since it crucially relies on the fact that the ranking for the particular language is already given, although the ranking of constraints has to be learned as well.[5]

6.3 Comparison to Inconsistency detection and surgery

Inconsistency detection and surgery (Tesar et al. 2003) makes use of paradigmatic comparison to determine underlying forms. As a first step, the learner tries to find a ranking for the language data. If there is no such ranking, meaning that there is inconsistency in the data, she will modify her lexicon. To be able to do so she will gather paradigmatic information. Forms that do not show any alternation in the paradigm are faithfully mapped onto underlying forms. All alternating forms are listed as unspecified. She will then modify one of the unspecified forms and try again to find a ranking for the data. If she does not find one, she will reset the modified form to unspecified and try to modify another form she listed as unspecified. She will proceed with modifying lexicon and ranking in turns until she finds a lexicon consistent

[5] Tesar & Smolensky (2000) state that the "lexicon learner" has to function "in concert" with the "grammar learner" (p. 80), meaning that the learning of underlying forms through Lexicon Optimization has to proceed interactively with grammar learning through constraint demotion. The present paper obliges by giving the formal account.

with a ranking. Via *surgery* the learner will be able to remember which forms have already been tested. This algorithm proceeds in an off-line fashion: when no ranking is found that is consistent with all the data, the lexicon is modified, after gathering all possible surface forms. But a learner does not know when she gathered enough data to go on with learning.

6.4 Considering multiple grammars

Another approach to the learning of underlying forms is considering multiple grammars (in fact, *all* grammars possible) at a time, as in Jarosz (2006). It makes use of the Expectation Maximization Algorithm (Dempsey et al. 1977) applied to Optimality Theoretic grammars. Within probabilistic unsupervised learning, all possible underlying forms and all possible rankings of constraints are initially equally probable. The probability of the underlying forms and the probability of the rankings are computed in combination with the probability of observed forms.

In this approach, the probability of all possible constraint rankings and underlying forms is computed, consulting the distribution of surface forms iteratively. Every iteration step consults all possible grammars. This means that, at all times in learning, all constraint rankings (i.e., all possible grammars *N!*) are present, and that, at all times in learning, all surface forms are present, and that, at all times in learning, all possible underlying forms are present, i.e. stored in the lexicon. This does not scale well: a constraint set of e.g. 30 constraints yields 30! possible grammars… While this might be a proper mathematical model for finding rankings and underlying forms given a distribution of surface forms, it is not suitable as a learnability approach.

7 Discussion

The alternatives discussed in 6.2-6.4 have in common that they are all *off-line* learning approaches: the learner first has to gather paradigmatic information before she can begin to modify her lexicon. This is not a very natural approach to language acquisition. It is not clear at what point the learner knows that she gathered enough information and will not encounter any further alternations. Moreover, she has to maintain access to all the observable forms or even to all possible grammars at all times in learning. This implies unlikely mnemonic processes.

In this paper, it has been shown how underlying forms can be learned effectively by a rather stupid on-line learning algorithm that takes meaning into account: on-line learning of underlying forms. An on-line approach of learning is better than an off-line approach, because one form is processed at a time, under one ranking at a time. Former processed forms or rankings do not have to be remembered, because their occurrence is implicitly stored in the ranking of the constraints. The ranking is adjusted systematically. No extra learning mechanisms are required than the ones already involved in a general grammar learning model of OT (e.g. Boersma 1997): interpretation of incoming forms and constraint reranking as a reaction to error detection is enough. The learning of underlying forms goes hand in hand with learning the grammar. This resolves the learning problem whether first the grammar or the lexical representations have to be learned.

The learning approach of underlying forms makes use of grammatical restrictions on the lexicon in form of lexical constraints. This constitutes in effect a partial grammaticalization of the lexicon.

References

Boersma, P. (1997). How we learn variation, optionality, and probability. *IFA Proceedings* 21:43-58.

- (2005). Some listener-oriented accounts of hache-aspiré in French. ROA 730.
- (2006a). The acquisition and evolution of faithfulness rankings. Handout 14th Manchester Phonology Meeting, May 2006.
- (2006b). A programme for bidirectional phonology and phonetics and their acquisition and evolution. To be made available on ROA.

Boersma, P. & D.d Weenink (1992-2006). *Praat: doing phonetics by computer (Version 4.4)*. [Computer program]. Retrieved from www.praat.org.

Dempster, A.P., N.M. Laird, and D.B. Rubin (1977). Maximum likelihood from incomplete data via the EM Algorithm. *Journal of Royal Statistics Society* 39(B):1-38.

Itô, J, A. Mester & J. Padgett (1995). Licensing and underspecification in Optimality Theory. *Linguistic Inquiry* 26:571-613.

Jarosz, G. (2006). Probabilistic unsupervised learning of Optimality Theoretic Grammars. Handout 3rd Old World Conference in Phonology, Budapest, January 2006.

McCarthy, J. & A. Prince (1993). Prosodic Morphology I: constraint interaction and satisfaction. *Technical Report* 3, Rutgers University Center of Cognitive Science. ROA 482.

Prince, A. & P. Smolensky (1993). Optimality Theory: Constraint interaction in generative grammar. *Technical Report Ru CCS TR-2*. Rutgers Center for Cognitive Science.

Tesar, B. (1995). Computational Optimality Theory. Diss., Department of Computer Science, University of Colorado, Boulder. ROA 90.

Tesar, B. & P. Smolensky (1996). Learnability in Optimality Theory (long version). *Technical Report JHU-ogci-96-4*. Department of Cognitive Science, Johns Hopkins University, Baltimore.

- (2000). Learnability in Optimality Theory. MIT Press, Cambridge, Mass.

Tesar, B., J. Alderete, G. Horwood, N. Merchant, K. Nishitani & A. Prince (2003). Surgery in language learning. *Proceedings of WCCFL 22*, pp. 477-490. ROA 619.

Wexler, K. & P. Culicover (1980). *Formal principles of language acquisition*. Cambridge, MA: MIT Press.