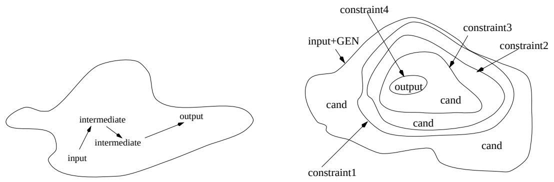
On Reversing the Generation Process in Optimality Theory

J. Eric Fosler
U.C. Berkeley and International Computer Science Institute
1947 Center Street, Suite 600, Berkeley, CA 94704
fosler@icsi.berkeley.edu

Abstract

Optimality Theory, a constraint-based phonology and morphology paradigm, has allowed linguists to make elegant analyses of many phenomena, including infixation and reduplication. In this work-in-progress, we build on the work of Ellison (1994) to investigate the possibility of using OT as a parsing tool that derives underlying forms from surface forms.



a. Derivational Phonology b. Optimality Theory

Figure 1: Search Spaces Within Different Paradigms

1 Introduction

Optimality Theory (Prince and Smolensky, 1993) is a constraint-based phonological and morphological system that allows violable constraints in deriving output surface forms from underlying forms. In OT a system of constraints selects an "optimal" surface output from a set of candidates. The methodology allows succinct analyses of phenomena such as infixation and reduplication that were difficult to describe under sets of transformational rules.

Several computational methods for OT have been produced within the short amount of time since Prince and Smolensky's paper (Ellison, 1994; Tesar, 1995; Hammond, 1995). These systems were designed as generation systems, deriving surface forms from an underlying lexicon. There have, however, been no computational models of OT parsers that derive underlying forms from the surface form.[1] In this work, we lay the theoretical groundwork for using OT as a parsing tool.

2 Comparing Derivational Methods to Optimality Theory

In traditional computational phonology/morphology systems such as two-level phonology (Koskenniemi, 1983), grammars that generate surface forms are invertible, allowing parsing back into underlying forms. In a derivational framework, the grammar converts underlying forms to surface outputs via transformations; the input and output share the same space (Figure 1a). In the one-level version of OT that most computational methods use, the space is populated with candidate outputs created by a generator function GEN operating on input strings. The search narrows in on an optimal output (Figure 1b) using evaluation constraints in a process called EVAL; successively smaller boundaries are cut out by the constraints until only one candidate remains. It is easy to see why the derivational method can be run backward: it just retraces derivational links in the graph. It is not obvious, though, how the input can be found from the search space in OT.

[1]Some of the computational work in OT confusingly uses the term "parsing" to refer to generation.

3 Tagalog Infixation

Infixation has traditionally been a difficult problem for computational models that use two-level phonology (Sproat, 1992). Infixation in Tagalog, however, has been modeled using OT (McCarthy and Prince, 1995). In Tagalog, the *um* affix can appear as a prefix, or "move" slightly into the word to which it is attaching (French, 1988).

Root	with *um*	Gloss
alis	um-alis	"leave"
sulat	s-um-ulat	"write"
gradwet	gr-um-adwet	"graduate"

McCarthy and Prince analyze *um* as a prefix, which moves into a word to reduce the number of coda consonants. They postulate two competing constraints, ALIGN-PREFIX and the higher-ranked NOCODA. ALIGN-PREFIX states that the prefix should remain as close to the front of the word as possible. NOCODA penalizes syllables with coda consonants.

In the OT derivation of *grumadwet* from *um*+*gradwet* (Figure 2), the winning candidate violates NOCODA twice, while the first two candidates violate it three times. The final candidate is pruned since it violates the ALIGN constraint more times than the winner.

	Candidates	NoCoda	Align
	um.grad.wet	* * *!	
	gum.rad.wet	* * *!	*
√	gru.mad.wet	**	**
	gra.dum.wet	**	* * *!*

Figure 2: OT Evaluation for Tagalog Infixation

```
(Morpheme Structure)  ( PPWWWWWWW ) | ( WPPWWWWWW ) | ( WWPPWWWWW )
 (Syllable Structure) ( NCOONCONC ) | ( ONCONCONC ) | ( OONONCONC )
(Phoneme Structure)   ( umgradwet ) | ( gumradwet ) | ( grumadwet )
                          Cand1           Cand2           Cand3
```

Figure 3: Candidate outputs for um+gradwet in an FST

4 Ellison's Conversion Method

Ellison (1994) provides a paradigm for converting Optimality Theory constraints into Finite State Transducers. He requires that Eval constraints output binary marks when ranking candidates and be describable as a regular language; the output of Gen must also be describable by a regular language. As Ellison points out, most constraints can be reformulated to be binary. He is able to build FST representations for the constraints that he considers, showing them to be regular.

For the Tagalog example, Gen will output the regular language shown in Figure 3 for the first three candidates (*umgradwet*, *gumradwet*, and *grumadwet*).[2] Each candidate consists of segments associated with a syllable structure position and a morpheme structure marker.[3]

We now consider the Align-prefix constraint, restricting the prefix to occur as early in the word as possible. This is encoded as an FST that writes marks on an output "Harmony Marks" tape. A "1" is written for any word (W) morphological material that precedes prefix (P) material, and a "0" is written for any other segment.

```
   (Harmony Marks)  ( 1 *  0 +  0 * ) | ( 0 * )
(Morpheme Structure)( W    P    W   ) | ( W   )
 (Syllable Structure)( ●    ●    ●   ) | ( ●   )
 (Phoneme Structure)( ●    ●    ●   ) | ( ●   )
```

Figure 4: Align-prefix FST Regular Language

The regular language generated by this FST (Figure 4) has a very simple structure. Any Ws before Ps on the Morpheme Structure tape get a harmony violation mark. Taking the product of this language with the optimal candidate scores the candidate (Figure 5). The harmony marks include two non-harmonic marks (i.e. "1"s); in the OT tableau in Figure 2, we see that Align also gives two marks to the optimal candidate.

We can encode a similar FST for NoCoda. This FST examines the syllable structure tape to give harmony marks (Figure 6)— codas (Cs) get a harmony violation mark, onsets (O) and nuclei (N) are unmarked. As in the OT tableau, the winning candidate (Figure 7) violates NoCoda twice.

[2] For brevity, we are not considering other candidates.

[3] We have extended Ellison's work by adding a third tape that marks segments as belonging to the prefix or to the word.

```
    (Harmony Marks)  ( 1 1 0 0 0 0 0 0 0 )
(Morpheme Structure) ( W W P P W W W W W )
 (Syllable Structure) ( O O N O N C O N C )
 (Phoneme Structure) ( g r u m a d w e t )
```

Figure 5: Scoring of grumadwet by Align-prefix

```
    (Harmony Marks)  ( 1 | 0 | 0 )*
(Morpheme Structure) ( ● | ● | ● )
 (Syllable Structure) ( O | N | C )
 (Phoneme Structure) ( ● | ● | ● )
```

Figure 6: Regular Language generated by NoCoda

Once the OT constraints are represented as FSTs, combining all of the Eval constraints into one transducer is a straightforward product. Ellison augments the product procedure so that harmony marks are concatenated by the resulting transducer.

We have used two different types of harmony marks in the Align-prefix and NoCoda FSTs, representing the ranking of the two rules as suggested by McCarthy and Prince. The higher-ranked NoCoda constraint outputs "2" marks while Align-prefix outputs "1" marks.[4] Harmonic comparisons between the candidates will consider the candidates with the smallest number of "2" marks first, followed by the smallest number of "1" marks. Marks are not added together, rather, the count of each type of mark is the deciding factor in evaluation.[5]

The output of Gen and the constraints of Eval are combined into a single transducer by taking the product of all of the FSTs. For the Tagalog example, the output rankings for the candidates are shown in Figure 8. Using the harmonic marks to prune the resulting transducer reveals the optimal candidate (Figure 9).

5 Extensions to Parsing

Ellison's approach gives us an elegant method of performing OT generation using finite state automata. Nevertheless, the system cannot parse the output string back into underlying surface forms. In a derivational paradigm (Figure 1a) , the input and output forms are enclosed in the same space. The derivational grammar is a transform that one can invert using FSTs, searching for the input using the output.

Ellison's FSTs transform output candidates to harmony marks; even so, the inversion of these FSTs are useless. The crucial point is that Gen hides the surface-form-to-candidate mapping; in Ellison's system the Eval portion of the system only combines with the *output* of Gen, so the mapping is lost. For invertability it is critical that the FST have access to both input and output forms.

In the version of OT (one-level OT) Ellison incorporated into his system, outputs of Gen are constrained

[4] Ellison uses only one type of mark and determines rank ordering from the relative positions of marks for each output segment. These two methods are equivalent.

[5] One "2" is worse than two "1"s.

```
    (Harmony Marks)  ( 0 0 0 0 0 2 0 0 2 )
(Morpheme Structure) ( W W P P W W W W W )
 (Syllable Structure) ( O O N O N C O N C )
 (Phoneme Structure) ( g r u m a d w e t )
```

Figure 7: Scoring of grumadwet by NoCoda

```
(Harmony Marks)      002000000020000020 | 010020000020000020 | 010100000020000020
(Morpheme Structure) P P W W W W W W W  | W P P W W W W W W  | W W P P W W W W W
(Syllable Structure) N C O O N C O N C  | O N C O N C O N C  | O O N O N C O N C
(Phoneme Structure)  u m g r a d w e t  | g u m r a d w e t  | g r u m a d w e t
```

Figure 8: Output of OT-FST System

```
(Harmony Marks)      101000000002000002
(Morpheme Structure) W W P P W W W W W
(Syllable Structure) O O N O N C O N C
(Phoneme Structure)  g r u m a d w e t
```

Figure 9: Pruned Output of OT-FST System

to be similar to the input. McCarthy & Prince (1994) abandon this constraint principle, and use *faithfulness* constraints in EVAL to achieve the same effect within "modern" two-level OT. This will be a critical move for the OT-FST paradigm.

In two-level OT, GEN generates all strings; faithfulness constraints in EVAL minimize the inserted and deleted material between underlying and candidate surface forms. By specifically modeling the faithfulness constraints, we now allow the FST to have access to the input-output correspondences crucial for searching for underlying forms. The remaining question, however, is whether faithfulness constraints *can* be modeled by regular grammars. Several formulations of two-level OT faithfulness constraints are discussed by McCarthy and Prince (1994) and Orgun (1994). To illustrate the flavor of these constraints and how they might be regularizable, we consider two constraints, CORR and MATCH (named for their similarity to Orgun's constraints). For our Tagalog example, we add two tapes for the underlying word and prefix forms (Figure 10). The CORR constraint requires that for every element in the surface phoneme string there is a segment in the underlying word or prefix, and vice versa. MATCH constrains the surface string phoneme to match[6] those in the word and prefix, and vice versa (Figure 11). Using these constraints, the OT-FSTs should be able to generate and parse in the Tagalog example.

```
(Morpheme Structure)          W W P P W W W W W
(Syllable Structure)          O O N O N C O N C
(Surface Phoneme Structure)   g r u m a d w e t

(Word Phonemes)               (g r a d w e t)
(Prefix Phonemes)             (u m)
```

Figure 10: Adding Word and Prefix Tapes

The additional computational complexity for implementing this type of system may be quite large; the search space for determining unknown strings at parse time will make for a slow implementation unless suitable heuristics are found for searching over each type of string. Systems of this type are likely to become even more complex as more information such as moraic structure is added. We envision that these heuristics will be based on the harmony mark scoring of the FST, but the exact nature of this is left to future work.

6 Conclusions & Future Work

Current Computational Optimality Theory systems provide solutions for OT generation, but deriving underlying forms from surface forms is not possible within these

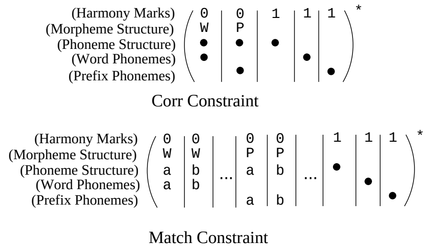


Figure 11: Faithfulness Constraints

systems. In order to extend any generation system to an OT parsing system, two-level Optimality Theory should be a critical component, since it moves the hidden relationship between input and output out of GEN and into EVAL. With two-level OT, the mapping from input to output can be directly operated upon by computational theories.

We have proposed using two-level OT to extend Ellison's technique for representing constraints as finite state transducers. By explicitly representing the input-to-output mapping using two-level OT, we have laid the theoretical groundwork for recovering underlying forms from surface forms.

In future work, we will implement the extensions to Ellison's algorithm allowing us to morphologically analyze cases like the Tagalog example. Search complexity will, however, be an issue in the implementation of the system; after an initial brute-force implementation, work must be focused on determining how the harmony marks can be used to heuristically guide the parser search.

Acknowledgments

We would like to thank Dan Jurafsky, Orhan Orgun, Sharon Inkelas, Nelson Morgan, Su-Lin Wu, and three anonymous ACL reviewers for comments, suggestions, and support.

References

T.M. Ellison. Phonological derivation in optimality theory. In *COLING-94*, 1994.

K. French. *Insights into Tagalog: Reduplication, Infixation, and Stress from Nonlinear Phonology.* M.A. Thesis, Summer Institute of Linguistics and University of Texas, Arlington, 1988.

M. Hammond. Syllable parsing in English and French. Rutgers Optimality Archive, 1995.

L. Karttunen. Kimmo: A general morphological processor. In *Texas Linguistics Forum 22*, 1983.

K. Koskenniemi. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production.* Ph.D. thesis, University of Helsinki, 1983.

J. McCarthy and A. Prince. Prosodic morphology, parts 1 and 2. Prosodic Morphology Workshop, OTS, Utrecht, 1994.

J. McCarthy and A. Prince. Prosodic morphology. In J. Goldsmith, editor, *Handbook of Phonological Theory*, pages 318–366. Basil Blackwell Ltd., 1995.

O. Orgun. Containment: Why and why not. Unpublished ms., U. of California-Berkeley, Department of Linguistics, July 1994.

A. Prince and P. Smolensky. Optimality theory. Unpublished ms., Rutgers University, 1993.

R. Sproat. *Morphology and Computation.* MIT Press, Cambridge, MA, 1992.

B. Tesar. *Computational Optimality Theory.* Ph.D. Thesis, U. of Colorado-Boulder, Department of Computer Science, 1995.

[6]Here we mean *be identical to*; this definition can be extended with features and underspecified elements.