T-ORDERS

Arto Anttila and Curtis Andrus
Stanford University

Abstract

In Optimality Theory, a set of grammatical constraints defines a space of possible languages. This space is called the FACTORIAL TYPOLOGY. In this paper, we note that every factorial typology defines a T-ORDER, i.e. a set of implicational universals that hold among <input, output> pairs. T-ORDER GENERATOR is a Windows program that generates t-orders. The input to the program is a factorial typology; the output is a t-order visualized as a directed graph. This structure has a useful application in the study of variation: it imposes universal limits on the quantitative variation permitted by a constraint set. These limits hold under several theories of variation, including Multiple Grammars, Partially Ordered Grammars, and Stochastic Optimality Theory.

1. A simple example

In Optimality Theory (Prince and Smolensky 1993/2004), a set of grammatical constraints defines a space of possible languages. This space is called the FACTORIAL TYPOLOGY. The structure of factorial typologies is linguistically interesting and worth exploring in detail. We start by illustrating this with a simple example from English phonology. However, our point is a general one and similar examples could easily be drawn from other languages as well as from other subfields of linguistics (morphology, syntax, semantics).

In many dialects of English, word-final *t,d* is variably deleted, as shown in (1):

(1)	It cost ~ cos' five dollars.	(*t* before a consonant)
	It cost ~ cos' us five dollars.	(*t* before a vowel)
	That's how much it cost ~ cos'.	(*t* before a pause)

Whether *t,d*-deletion applies or not depends on various factors, in particular the quality of the following segment. It is well known that *t,d*-deletion is more common before consonants than before pauses or vowels (see Coetzee 2004:218 and references there). Typically, *t,d*-deletion applies variably in all three environments depending on the speaker, speech rate, lexical item, etc., but the quantitative tendency is robust and clear. This is shown by the cross-dialectal data in (2).

(2) *t,d*-deletion data from five dialects (Coetzee 2004: 218)

		_C	_V	_##
Chicano English (Los Angeles)	*n*	3,693	1,574	1,024
(Santa Ana 1991:76, 1996:66)	% deleted	62	45	37
Tejano English (San Antonio)	*n*	1,738	974	564
(Bayley 1995:310)	% deleted	62	25	46
AAE (Washington, DC)	*n*	143	202	37
(Fasold 1972:76)	% deleted	76	29	73
Jamaican mesolect (Kingston)	*n*	1,252	793	252
(Patrick 1991:181)	% deleted	85	63	71
Trinidadian acrolect	*n*	22	43	16
(Kang 1994:157)	% deleted	81	21	31
Neu data	*n*	814	495	--
(Neu 1980:45)	% deleted	36	16	--

If we look at each dialect individually, we see three different percentages, e.g. 62-45-37. If we look across several dialects, we observe a pattern: the first number is always higher than the second and the third whereas the latter two can occur in either order. The generalization is summarized in (3):

(3) (a) In all dialects, deletion rate is highest in _C.
(b) Deletion rates in _V and _## may occur in either order.

This example illustrates a general point. There are two types of quantitative patterns in languages: QUANTITATIVE UNIVERSALS that hold true across languages, and QUANTITATIVE PARTICULARS that are subject to cross-linguistic variation. This distinction should fall out from phonological theory. In this case, a satisfactory analysis should rule out hypothetical dialects where deletion rate is higher in _{V, ##} than in _C, while allowing dialects where deletion rate is higher in _V than in _## as well as dialects that have the opposite pattern. In what follows, we will show how these two types of quantitative patterns emerge from Optimality Theory.

2. An optimality-theoretic analysis

Optimality Theory (OT, Prince and Smolensky 1993/2004) is a theory of constraint interaction in generative grammar. For a textbook exposition of the theory, see e.g. Kager 1999. At the heart of Optimality Theory there are three assumptions: (i) grammars of natural languages consist of constraints that make potentially conflicting structural demands; (ii) conflicts among constraints are resolved by strict ranking; (iii) constraints are universal, rankings are language-specific.

We now outline an optimality-theoretic analysis of *t,d*-deletion due to Kiparsky (1993). For alternative analyses, see Reynolds 1994:119-141 and Coetzee 2004:214-329. The analysis builds on five universal constraints:

(4)	*COMPLEX	Avoid consonant clusters within a syllable.
	ONSET	Syllables have onsets.
	PARSE	Segments belong to syllables.
	ALIGN-LEFT-WORD	Syllables cannot straddle word boundaries.
	ALIGN-RIGHT-PHRASE	Phrase-final consonants are also syllable-final.

The tableau in (5) shows one possible ranking of these five constraints. The assumption is that *t* is deleted if it is not parsed as part of a syllable. Syllables are marked by square brackets. This particular ranking makes the following predictions: in the prevocalic environment *t* is resyllabified into the following word and hence retained; in the preconsonantal and prepausal environments *t* is deleted.

(5) Sample ranking. Winners: *cost us* (no deletion), *cos' me* (deletion), *cos'* (deletion)

INPUTS	OUTPUTS	*COMPLEX	ONSET	ALIGN-L-W	ALIGN-R-P	PARSE
cost us	(a) [cost][us]	*!	*			
	(b) [cos]t[us]		*!			*
	(c) → [cos][tus]			*		
cost me	(a) [cost][me]	*!				
	(b) → [cos]t[me]					*
	(c) [cos][tme]	*!		*		
cost	(a) [cost]	*!				
	(b) → [cos]t				*	*

3. t-orders

What kinds of dialects does the analysis predict to be possible and what kinds of dialects does it exclude as impossible? This can be figured out by computing the factorial typology of the five constraints with the aid of OTSOFT (Hayes, Tesar, and Zuraw 2003). The program considers all the 120 total rankings of the 5 constraints and works out the predicted output patterns (= dialects, languages) in each case. Only 6 distinct output patterns are found. This implies that several distinct total rankings produce the same output pattern. The factorial typology is shown in (6). *t,d*-deletion is highlighted in grey.

(6) Factorial typology

```
There were 6 different output patterns.

               Output #1    Output #2    Output #3    Output #4
/cost us/:     [cost][us]   [cos]t[us]   [cos]t[us]   [cos][tus]
/cost me/:     [cost][me]   [cos]t[me]   [cos]t[me]   [cost][me]
/cost/:        [cost]       [cost]       [cos]t       [cost]

               Output #5    Output #6
/cost us/:     [cos][tus]   [cos][tus]
/cost me/:     [cos]t[me]   [cos]t[me]
/cost/:        [cost]       [cos]t
```

An inspection of the factorial typology shows that only four types of deletion systems are predicted (Kiparsky 1993:4):

(7)	(a)	Deletion in _C	= Output #5
	(b)	Deletion in _{C, V}	= Output #2
	(c)	Deletion in _{C, pause}	= Output #6
	(d)	Deletion in _{C, V, pause}, i.e. everywhere	= Output #3

Absent are systems with deletion only before a vowel, only before a pause, or only before a vowel or pause. Such systems cannot be derived by any ranking of the five constraints. In other words, the following IMPLICATIONAL UNIVERSALS hold true of all the predicted output patterns:

(8) Two implicational universals

(a) If *t,d*-deletion occurs before a vowel, it also occurs before a consonant.
(b) If *t,d*-deletion occurs before a pause, it also occurs before a consonant.

Implicational universals are generalizations that hold true no matter how the constraints are ranked. However, as this example illustrates, implicational universals are not immediately obvious: the statements in (8) are not easy to read off the constraint definitions ((4)), the tableau ((5)), or the factorial typology ((6)). This raises the possibility that there may well be more hidden implicational universals lurking in the factorial typology and it might well be worth our while to figure them all out.

We start by restating the factorial typology in a slightly different way. As shown in (9), each cell in a factorial typology corresponds to an <input, output> pair. This allows us to view implicational universals as a relation among <input, output> pairs.

(9) Factorial typology as <input, output> pairs

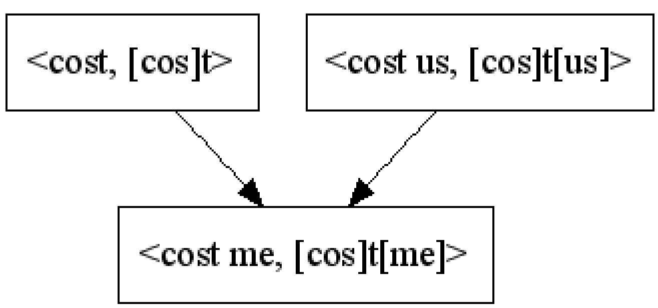
```
Pattern #1                Pattern #2                Pattern #3
</cost us/,[cost][us]>    </cost us/,[cos]t[us]>    </cost us/,[cos]t[us]>
</cost me/,[cost][me]>    </cost me/,[cos]t[me]>    </cost me/,[cos]t[me]>
</cost/,[cost]>           </cost/,[cost]>           </cost/,[cos]t>

Pattern #4                Pattern #5                Pattern #6
</cost us/,[cos][tus]>    </cost us/,[cos][tus]>    </cost us/,[cos][tus]>
</cost me/,[cost][me]>    </cost me/,[cos]t[me]>    </cost me/,[cos]t[me]>
</cost/,[cost]>           </cost/,[cost]>           </cost/,[cos]t>
```

We can now express the two implicational universals as pairs of <input, output> pairs, as in (10), or visualize them as a directed graph, as in (11).

(10) Implicational universals as pairs of <input, output> pairs

(a) `<cost us, [cos]t[us]> --> <cost me, [cos]t[me]>`
(b) `<cost, [cos]t> --> <cost me, [cos]t[me]>`

(11) Implicational universals as a directed graph

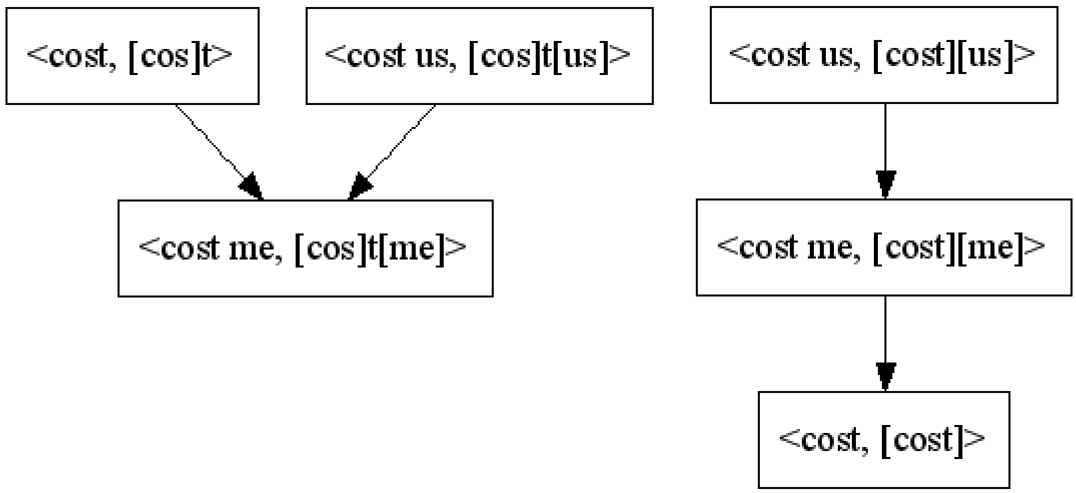


We call the set of all implicational universals in a factorial typology a T-ORDER (Anttila, Fong, Benus, and Nycz 2006). We can construct the t-order by looking up all the distinct <input, output> pairs that made it into the factorial typology, by constructing all the possible pairs of such pairs, and by checking which <input, output> pairs entail which other <input, output> pairs for all the output patterns in the factorial typology. In our example, the resulting t-order consists of five implicational universals in (12), visualized as the two disjoint graphs in (13).

(12) t-order as pairs of <input, output> pairs

(a) `<cost us, [cos]t[us]> --> <cost me, [cos]t[me]>`
(b) `<cost, [cos]t> --> <cost me, [cos]t[me]>`
(c) `<cost us, [cost][us]> --> <cost me, [cost][me]>`
(d) `<cost me, [cost][me]> --> <cost, [cost]>`
(e) `<cost us, [cost][us]> --> <cost, [cost]>`

(13) t-order as a directed graph



The right hand graph reveals three additional implicational universals: if syllabification is crisp (i.e. no stray *t*, no resyllabification) before vowels, the same should hold before consonants and pauses, and if syllabification is crisp before consonants, the same should hold before pauses. This shows that even a small factorial typology may hide several unobvious implicational universals.

4. The T-Order Generator

The problem with factorial typologies is that they are hard for humans to understand. OTSOFT does a good job computing factorial typologies, but the result is usually too large and too complex to make immediate sense. Even if the factorial typology is small as in (6), figuring out the t-order with paper and pencil is a tedious exercise. In analyses of realistic size, the number of output dialects may well run in the hundreds. In such cases, the only viable option is to use a computer to work out the <input, output> pairs and the implicational universals among them. This can be done by the T-Order Generator.

The T-Order Generator is a computer program that takes a factorial typology as input and returns the corresponding t-order as a directed graph.

5. An application to quantitative variation

The left hand graph in (13) shows that the grammar captures the empirical asymmetry in *t,d*-deletion: the preconsonantal environment is more favorable to deletion than the other two environments. However, more needs to be said: all the dialects in (2) are variable and the implicational universal surfaces quantitatively, not categorically. The question is how to relate the t-order to variable/quantitative patterns. For the purposes of this illustration, let us follow Kiparsky (1993) and adopt the Multiple Grammars Theory of variation. Its basic assumptions are stated in (14).

(14) The Multiple Grammars Theory of variation (Kiparsky 1993, Anttila to appear):

(a) Variation arises from multiple grammars within/across individuals.
(b) The number of grammars predicting an output is proportional to the frequency of occurrence of this output.

Assume an individual whose grammar consists of three total rankings that generate Outputs #1, #5, and #6 in the factorial typology. At the moment of speaking, the individual selects a grammar at random. In the long run, the following quantitative pattern will emerge: no deletion before vowels, 67% deletion before consonants, and 33% deletion before pauses.

(15) Sample grammar: {#1, #5, #6}

	Output #1	Output #5	Output #6	Del. rate
/cost us/:	[cost][us]	[cos][tus]	[cos][tus]	0/3
/cost me/:	[cost][me]	[cos]t[me]	[cos]t[me]	2/3
/cost/:	[cost]	[cost]	[cos]t	1/3

The crucial observation is this: the implicational universals guarantee that it is impossible to construct a grammar that would predict more *t,d*-deletion before vowels or pauses than before consonants. Hence the following quantitative universal holds:

(16) Quantitative universal: *t,d*-deletion rate before consonants is at least as high as *t,d*-deletion rate before vowels or pauses.

In contrast, no implicational universal connects the vowel and pause environments. In Output #2, we find deletion before a vowel, but not before a pause; in Output #6 we find deletion before a pause, but not before a vowel. This means that the two environments can occur in either order quantitatively as well. This corresponds to the quantitative observations.

In sum, we have seen that t-orders play an important role in variation: they impose universal limits on the quantitative variation patterns permitted by a constraint set. We chose to illustrate this in terms of the Multiple Grammars Theory (Kiparsky 1993), but we could just as well have chosen Partially Ordered Grammars (Anttila and Cho 1998) or Stochastic Optimality Theory (Boersma and Hayes 2001). This is because t-orders follow from the factorial typology and in all these theories the factorial typology is the same.

6. Summary and future directions

Every optimality-theoretic grammar defines a t-order. We conclude by listing some reasons why t-orders are linguistically interesting and why it seems worthwhile to work them out:

(17) Why are t-orders linguistically interesting?

(a) t-orders spell out the universal predictions of a constraint set, irrespective of how the constraints are ranked

(b) t-orders are predicted to hold true both qualitatively in invariant systems and quantitatively in variable systems

(c) t-orders are a consequence of OT, not a new theoretical device that one might or might not want to adopt

(d) t-orders have general validity: they hold true under several theories of variation, e.g. Multiple Grammars (Kiparsky 1993), Partially Ordered Grammars (Anttila and Cho 1998), and Stochastic OT (Boersma and Hayes 2001)

(e) t-orders serve as a diagnostic for the adequacy of constraints: if the t-order conflicts with the empirical typology, either qualitatively or quantitatively, no amount of ranking will help – new constraints will be necessary

(f) t-orders are universal and therefore do not have to be learned

We introduced t-orders through a simple example of phonological variation in English. Further illustrations can be found in the following case studies: consonant clusters in Singapore English (Anttila, Fong, Benus, and Nycz 2005), metrical variation in Finnish, and word order variation in English (Anttila 2006). Another empirical domain where t-orders have an obvious application is semantic ambiguity and partial blocking (Anttila and Fong 2000, 2004).

In this paper, we constructed t-orders from factorial typologies. It should be possible to construct t-orders directly from constraints, candidates and violation marks. How exactly this should be done remains an open question.

7. About the program

T-Order Generator was programmed by Curtis Andrus in the Python programming language during the Winter Quarter of 2006. T-Order Generator can be downloaded from

http://www.stanford.edu/~anttila/research/torders/torder.zip

The present version of T-Order Generator reads factorial typology files produced by OTSoft (Hayes, Tesar, and Zuraw 2003). When running OTSoft, choose the options **Include Ranking Arguments, Assume Transitivity of Domination** and **Include Illustrative Minitableaux**. These are essential for the T-Order Generator. OTSoft can be downloaded from

http://www.linguistics.ucla.edu/people/hayes/otsoft/

T-Order Generator requires Graphviz software to be installed in order to draw t-order graphs. More information on Graphviz software can be found at

http://www.graphviz.org/

After downloading T-Order Generator, take the following steps.

- Unzip the program file.
- Run the program by clicking on tordergui.exe.
- Open the factorial typology file conventionally named MYFILEDraftOutput.txt.
- Select the desired user options (see below) and click on **Generate T-Order**.

The program allows the user to choose the following options:

Save T-Order: Save the t-order as a text file.

Save Graph: Save the t-order into a graph file readable with Graphviz software. The program can output the t-order graph to any supported file format.

Save Pair Possibilities: Write all possible <input, output> pairs into a text file.

Collapse Cycles: Remove cycles, i.e. <input, output> pairs that imply each other, and collapse the nodes into a box. The program computes the transitive reduction (i.e. the t-order with all unnecessary edges removed) using the property that the t-order is the transitive closure of a relation (i.e. all edges implied by transitivity are in the t-order). While the t-order is the transitive closure of a relation, the final graph that the program outputs has been simplified and no longer has this property.

Split Disjoint Graphs: Split the t-order graph into disjoint components and output each component to a separate file. This is useful when the graphs get large. The program names the separate graph files by taking the filename given to it by the user and by attaching a number to the end. This is good to keep in mind in order to avoid overwriting existing files. For example, if you previously had a graph named *test1.jpg* and then tried to output a graph named *test* split into two different graphs, the program would create files *test0.jpg* and *test1.jpg* and the original file *test1.jpg* would be overwritten.

References

Anttila, Arto. 2006. 'Variation in Optimality Theory' (4 handouts), Phonology Fest, Indiana University, Bloomington, June 19-22, 2006. http://www.stanford.edu/~anttila/teaching/indiana/variation.html

Anttila, Arto. to appear. 'Variation and optionality', in Paul de Lacy (ed.), *The Cambridge Handbook of Phonology*, Cambridge University Press, Cambridge.

Anttila, Arto, and Young-mee Yu Cho. 1998. 'Variation and change in Optimality Theory', *Lingua* 104, 31-56, Special issue on Conflicting Constraints.

Anttila, Arto and Vivienne Fong. 2000. 'The Partitive Constraint in Optimality Theory', *Journal of Semantics* 17, 281-314.

Anttila, Arto and Vivienne Fong. 2004. 'Variation, ambiguity, and noun classes in English', *Lingua* 114, 1253-1290. Also at http://www.stanford.edu/~anttila/.

Anttila, Arto, Vivienne Fong, Stefan Benus, and Jennifer Nycz. 2005. 'Consonant clusters in Singapore English', Ms., Stanford University, Columbia University, and New York University.
Boersma, Paul, and Bruce Hayes. 2001. 'Empirical tests of the Gradual Learning Algorithm', *Linguistic Inquiry* 32, 45-86.
Coetzee, Andries. 2004. *What it Means to be a Loser: Non-optimal Candidates in Optimality Theory*, Ph.D. dissertation, University of Massachusetts, Amherst.
Hayes, Bruce, Bruce Tesar, and Kie Zuraw. 2003. OTSoft 2.1, software package. Available at http://www.linguistics.ucla.edu/people/hayes/otsoft/
Kager, René. 1999. *Optimality Theory*. Cambridge University Press, Cambridge, U.K.
Kiparsky, Paul. 1993. 'An OT Perspective on phonological variation'. Handout from Rutgers Optimality Workshop 1993, also presented at NWAVE 1994, Stanford University. Available at http://www.stanford.edu/~kiparsky/Papers/nwave94.pdf
Prince, Alan and Paul Smolensky 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*, Blackwell Publishing, Malden, Massachusetts.
Reynolds, William T. 1994. *Variation and Phonological Theory*. Ph.D. dissertation, Department of Linguistics, University of Pennsylvania.

Arto Anttila
Department of Linguistics
Stanford University
anttila@stanford.edu

Curtis Andrus
Department of Mathematics
Stanford University
candrus@stanford.edu