Learning with Properties: Restrictiveness and Typological Structure

Natalie DelBusso
Rutgers University

1 Introduction

A learner's task is to find the most restrictive grammar consistent with the data of their language. Languages in a typology differ in degree of restrictiveness: in a more restrictive language, some contrast is neutralized in optimal forms, while it is maintained in a less restrictive language. A learner of a more restrictive language faces the 'subset problem' of ruling out the superset grammar despite the data of a subset language being consistent with that of the superset language.

In Optimality Theory (OT; Prince & Smolensky 1993/2004), markedness constraints (M), violated by certain features or structures, favor neutralization, while faithfulness constraints (F), violated by disparities between inputs and outputs, favor contrast by preserving underlying feature values. This results in a long-recognized correlation between M>F rankings and degree of restrictiveness. The Output-Driven Learner algorithm (ODL; Tesar (2014) uses this correlation in approaching restrictiveness through Biased Constraint Demotion (BCD; Prince & Tesar 2006). BCD builds a bias into Recursive Constraint Demotion (RCD; Tesar 1995) that favors low ranking of faithfulness constraints. This implicitly encodes M>F but only imposes a ranking bias and does not encode the ordering in an Elementary Ranking Condition (ERC[1]). In this way, BCD avoids committing the learner to ranking information that does not come directly from the data. However, without an ERC representation, the faithfulness-low information cannot be used to detect inconsistency, a key piece part of the learning algorithm.

The learner can make gains in efficiency and restrictiveness by adopting certain M>F rankings as ERCs at the onset of learning. This is similar to Smolensky's (1996) proposal that in the initial state structural markedness constraints rank above faithfulness constraints. However, within a typology, rankings aligning with restrictiveness involve only *particular* markedness and faithfulness constraints rather than blocks thereof. An OT grammar is a set of constraint rankings; not all constraints are crucially ranked relative to all others in grammars. Instead, there is a set of core rankings and their interactions that define the grammars. Property Theory (PT; Alber & Prince in prep., Alber, DelBusso & Prince 2016, DelBusso 2018) identifies this key set of rankings that distinguish the grammars of a typology, and is also the learner's target set.

This paper modifies Tesar's (2014) ODL by incorporating typological-level information from Property Analysis. In Property-ODL (PODL), the learner begins with a set of set of M>F ERCs taken from property values (PERCs), encoding restrictiveness more explicitly than BCD. These rankings are, however, defeasible, and can be retracted in the first learning stage if inconsistent with the data. PODL was tested in a learning simulation for the phonological system in Tesar (2014), Paka, which presents the challenging case of languages in paradigmatic subset relations. In ODL, these require additional methods to be learned. PERCs eliminate the need for these in learning the paradigmatic subsets and overall reduce the use of the less-tested methods Fewest Set Features and Max-Mismatch Ranking in learning the grammars of the typology.

* Many thanks to Bruce Tesar, whose work, discussion, and feedback were essential to the development of this project, and to Alan Prince for discussion of the ideas.

[1] ERC (Prince 2002): three-valued vector recording constraint (C) preferences for a comparison of candidates: W = C prefers the winner; L = C prefers the loser; e = C does not distinguish the candidates.

Proceedings of *AMP 2019*

2 Restrictiveness and ODL

2.1 *BCD and restrictiveness in grammars* A grammar G1 is more restrictive than grammar G2 if G1 maps the same set of inputs to a subset of the outputs of G2 (Tesar 2014:306). For example, G1 might map both voiced and voiceless input segments to voiceless outputs, while G2 faithfully realizes contrastive voicing features. All G1 data is consistent with G2 as well.

Learning algorithms in OT including ODL enforce restrictiveness by exploiting the correlation between restrictive mappings and M (markedness) > F (faithfulness) rankings. For ODL, the main such method is BCD (Prince & Tesar 2006), which builds an M>F bias into RCD. RCD constructs a hierarchy, if one exists, where all the desired winners are optimal. It ranks every constraint as high as possible, regardless of the M/F distinction. BCD maintains this high ranking for M constraints, but reverses it for F constraints, placing them as low as possible.

RCD takes an ERC set and recursively iterates through two steps: i) identify all constraints that can be placed in the (next) highest stratum (those that favor no losers) and rank them there; ii) identify all ERCs that are satisfied or resolved by this ranking, and remove them from the current list. The steps repeat over increasingly smaller subsets of ERCs until none remain, as all are satisfied.

An important property of RCD is inconsistency detection. If, for a subset of ERCs, no constraint can be placed in the next stratum because all favor some loser(s), then no ranking can satisfy all remaining ERCs. A case of such inconsistency is shown in (1). Among the final three ERCs, there is no ranking of f2, m3 and m5 that satisfies all three; these ERCs form an unresolvable subset (URS), which is used in the PODL algorithm. Fusing the URS results in the ERC eeLLL, unsatisfiable under any ranking.[2]

(1) RCD inconsistency

	f1	m4	m5	f2	m3
ERC1	W			L	W
ERC2		W	L		L
ERC3			W	L	
ERC4				W	L
ERC5			L		W

BCD modifies RCD by ranking only the M constraints as high as possible, but favoring low ranking of Fs. If there is a choice between ranking an M and an F, BCD ranks M; if there is a choice between multiple Fs, BCD ranks the one that 'frees' the most Ms to be ranked in the next stratum. An F can thus be placed in a lower stratum even if it favors no losers, as for f1 in (2). The M>F ordering is, however, implicit: there is no ERC representing the ranking of m4>f1.

(2) BCD

	m4	f1	m5	f2	m3
ERC2	W		L		L
ERC1		W		L	W
ERC3			W	L	
ERC4				W	L

2.1 *Output-Driven Learner (ODL)* The Output-Driven Learner (ODL; Tesar 2014) simultaneously learns a grammar (rankings) and lexicon (underlying input feature values). To learn a grammar, ODL uses the error-driven method Multi-Recursive Constraint Demotion (MRCD; Tesar 1997), building and maintaining a support of ERCs that represents the current grammar hypothesis. Data forms are tested against this support; if the observed form is not the predicted winner, the learner constructs a winner-loser (WL) pair (the ERC resulting from the comparison of the observed form (the winner) and the incorrectly

[2] Fusion (Prince 2002, Brasoveanu & Prince 2011) is an operation on an ERC set that returns an ERC jointly entailed by the set. For each constraint column, fusion returns a value from {W,L,e} as follows: $L \circ X = L$; $e \circ X = X$; $X \circ X = X$, ($X \in \{W,L,e\}$).

predicted form (the loser)) and adds this pair to the support.

Along with learning a ranking, underlying feature values of the lexicon are set using inconsistency detection and paradigmatic comparison. For example, in Paka, if the learner has observed that both long and short vowels occur in optima and has added ranking information allowing such mappings, then a [-len] underlying value for a vowel that surfaces as [+len] is inconsistent. A language can be fully learned without all underlying feature values being set. Words are tested to see if values must be set by assessing whether the form differing from the observed one in the values of all unset features is correctly mapped to the observed output form under the current grammar. If so, no features need be set; if not, the learner tests each unset feature individually for setting.

There are four methods used in ODL: Phonotactic Learning (PL), Single-form learning (SF), Contrast Pairs (CP), and Fewest Set Features (FSF) (Tesar 2014:376 (8.23)). The learner has access to a representative set of the data at each stage and to morpheme identity, but not the underlying forms. PL takes place prior to morphological awareness. The learner begins with an empty support and the initial grammar hypothesis is the hierarchy generated by BCD, where all M constraints are in the first stratum and all Fs in the second. During PL, the learner learns F>M rankings, and assumes faithful mappings (the observed output = input). Following PL, in SF and CP stages the learner adds non-phonotactic ranking information and attempts to set underlying feature values where needed. If underlying feature values must but cannot be set using these methods—due to multiple consistent value settings—FSF is used to select the solution that involves setting the fewest feature values. The subset of methods used by the learner differs depending on the language being learned. Some languages require only PL, while others use all methods.

FSF aims to enforce restrictiveness in the lexicon. The method, used only when other learning steps and strategies are insufficient to learn a language, biases towards greater neutralization, as more inputs—those with any setting of the unset features—neutralize to the surface form. FSF crucially differs from other ODL methods in being an 'inductive' method that is not being directly driven by the data. For the Paka system (§3), FSF is needed for learning 2 languages that are paradigmatic subsets of others in the typology. However, FSF is not sufficient for learning all languages in another system that expands on Paka (Moyer 2017). For this system, Moyer (2017; Moyer & Tesar 2017, 2019) proposes another 'inductive' learning strategy, Max-Mismatch Ranking (MMR), in which the learner adds ranking information based on the hypothesis that any input must map to one of the observed outputs in their current grammar.

3 Paka and it Properties

3.1 *The Paka system* The Paka system derives interactions of stress and vowel length for a set of 16 2-syllable input words, built from all combinations of 4 roots (r1-r4) and 4 suffixes (s1-s4). Each morpheme is defined by the values of stress [±str] and vowel length [±len]. The set of 8 outputs is the subset of input forms that have exactly one stress. Candidates are represented using the notations in (3) to encode the feature values.

(3) Candidate notations

		Stress	
		+	-
Length	+	H	h
	-	X	x

CON consists of 4 markedness and 2 faithfulness constraints (4). m.ML and m.MR are violated by right or left stress position, respectively. m.NL is violated by any long vowel and m.WSP is violated by an long unstressed vowel. The faithfulness constraints are violated by unfaithful realization of stress or length feature values between input and corresponding output.

(4) CON

Constraint	One violation for:
m.ML	Right-aligned stress: *xX, xH, hX, hH
m.MR	Left-aligned stress: *Xx, Hx, Xh, Hh
m.NL	Long vowel: *h, H
m.WSP	Long unstressed vowel: *h
f.S	Unfaithful mapping of [±str]: *{X,H} ↔ {x,h}
f.L	Unfaithful mapping of [±len]: *{x,X} ↔ {h,H}

The Paka typology has 24 languages, evenly divided between by left- or right-aligned default stress position. Within each symmetric half, languages vary in presence/absence of long vowels and/or long unstressed vowels, and in whether stress position is uniform or varies depending on underlying features of inputs. The right-aligned half is shown in (5), using a representative subset of inputs (optima for all inputs shown in the Appendix); left-aligned counterparts differ in the relative ranking of m.MR and m.ML. Shading highlights key contrasts distinguishing the languages.

(5) Paka typology (right half)

Inputs	x.x	X.x	x.h	h.h	h.x	h.X
L1	x.X	x.X	x.X	x.X	x.X	x.X
L2	x.X	X.x	x.X	x.X	x.X	x.X
L3	x.X	x.X	x.H	x.H	x.X	x.X
L4	x.X	X.x	x.H	x.H	x.X	x.X
L5	x.X	x.X	x.H	h.H	h.X	h.X
L6	x.X	X.x	x.H	h.H	h.X	h.X
L7	x.X	X.x	x.H	x.H	H.x	x.X
L8	x.X	x.X	x.H	x.H	H.x	H.x
L9	x.X	X.x	x.H	x.H	H.x	H.x
L10	x.X	X.x	x.H	h.H	H.x	h.X
L11	x.X	x.X	x.H	h.H	H.x	H.x
L12	x.X	X.x	x.H	h.H	H.x	H.x

L1 is the most restrictive language: its phonotactic inventory includes a single form to which all inputs are mapped. In contrast, L12 faithfully realizes many underlying stress and length features and is less restrictive. Because of the one-stress-per-output requirement, some inputs cannot be faithfully mapped in any language. There are also super/subset relations between languages in the typology, both in terms of inventories (sets of optima) and morphological paradigms (mapping behavior). The latter, exemplified in the case of L7 and L8, presents a learning problem discussed more in §4.2.3 below.

3.2 *Property Analysis* A Property Analysis (PA) shows the internal structure of an OT typology by identifying the core constraint conflicts that define the grammars and how these interact. These ranking are the *properties* (Ps) that encode conflicts between two sets of constraints (the *antagonists*, X<>Y). The *values* of a property are the opposing rankings, α: X>Y and β: Y>X, each of which defines an ERC set. The set of possible P-value combinations for all Ps in a PA defines all and only the grammars of the typology as unique sets of values (Alber & Prince in prep., Alber, DelBusso & Prince 2016, DelBusso 2018).

The antagonists X and Y may be single constraints or classes of constraints with an operator *dom* or *sub* that picks out the dominant or subordinate member of the class, respectively (Alber & Prince in prep.). For example, {m.ML,m.MR}.dom refers to whichever of m.ML and m.MR is dominant in their relative ordering. When such a class is dominant, the P-value ERC is satisfied if either member of the class dominates the antagonist; when it is subordinate, the value ERC requires that the antagonist dominate all members (by transitive domination of the highest). The dual operator sub works similarly but designates the lowest ranked member of the set.

Properties are sometimes in interdependent relations. A *wide-scope* P is one for which every grammar

in the typology has a value, while a *narrow-scope* P is moot in some grammars where neither value is entailed (Alber & Prince in prep.). The scope of a narrow-scope P—the set of grammars having a value—is defined by value(s) of another P(s). This often aligns with empirical dependencies: for example, the choice of whether long vowels can be unstressed in a language's optima in Paka is only relevant among those languages with contrastive vowel length.

The PA of the Paka system consists of the 6 properties and their scopes shown in (6) and described below. The value table shows the PA characterization of the right-half grammars. Scope dependencies are represented in the *treeoid,* a graphic representation of the PA structure (Alber & Prince in prep.).[3]

(6) PA(Paka)

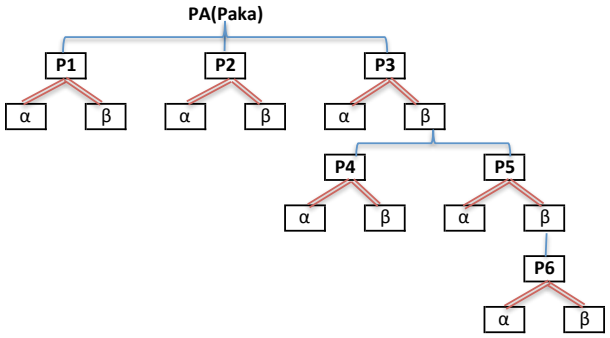
a. Properties

P		*Scope*	*Value ERCs*[4]	
P1	m.ML<>m.MR		WLeeee	LWeeee
P2	{m.ML,m.MR}.dom<>f.S		WWeeLe	LLeeWe
P3	m.NL<>f.L		eeWeeL	eeLeeW
P4	m.WSP<>f.L	P3β	eeeWeL	eeeLeW
P5	{m.ML,m.MR}.dom<>{m.WSP,f.L}.sub	P3β	WWeLee|WWeeeL	LLeWee,LLeeeW
P6	f.S<>{m.WSP,f.L}.sub	P5β	eeeLWe|eeeeWL	eeeWLe,eeeeLW

b. Value table (right half)

	P1	P2	P3	P4	P5	P6
L1	β	α	α			
L2	β	β	α			
L3	β	α	β	α	α	
L4	β	β	β	α	α	
L5	β	α	β	β	α	
L6	β	β	β	β	α	
L7	β	β	β	α	β	α
L8	β	α	β	α	β	β
L9	β	β	β	α	β	β
L10	β	β	β	β	β	α
L11	β	α	β	β	β	β
L12	β	β	β	β	β	β

c. Treeoid



P1 antagonizes the two stress-position M constraints and its values correlate with left and right default stress in a language. In P2, the dominant member of the set of m.ML and m.MR constraints conflicts with f.S, correlating with the presence of lexical stress when an underlying stressed segment is not in the default position. This trait can be seen in the input /X.x/ in the typology in (6) above: with P2α, default right stress is maintained, and with P2β, the output with left stress is optimal, satisfying f.S.

P3 and P4 values correlate with traits involving vowel length. P3 values characterize the difference between languages with no long vowels in outputs (α) and those with some long vowels (β). P4 values divide the latter set by presence (β) or absence (α) of unstressed long vowels in optima. P4's scope is defined by P3β: only if a language allows some long vowels is the choice encoded in P4 relevant.

P5 and P6 have complex antagonist sets and values align with choices involving the interaction of stress and length. In P5, the dominant stress position constraint conflicts with the subordinate member of the set {m.WSP, f.L} (determined by P4). Its values align with the choice between maintaining default stress position (α) or having length-driven stress (β) when there is an underlying long vowel in the non-default position (as in input /h.x/ above). Among languages where the heavy syllable is stressed (P5β), P6 makes a similar distinction when the other vowel is underlyingly stressed (as in input /h.X/).

The possible combinations of P1-P6 values distinguish the 24 grammars, as in the value table. These are the central rankings in the typology that determine the traits of the languages' optima.

[3] Some values combinations are ruled out by contradiction rather than scope: P2α is inconsistent with P5β+P6α.
[4] ERCs are written using the constraint order: m.ML-m.MR-m.NL-m.WSP-f.S-f.L.

4 PODL

PODL modifies ODL by incorporating typological information in the form of ERCs derived from Property Analysis. These PERCs are the values of MF properties set to M>F values and constitute the learner's initial grammar.

PERCs are full ERCs, not biases, and thus the ranking information they encode can be used in inconsistency detection at later learning stages, allowing for further gains of ranking and lexicon information during these stages. PERCs are also more targeted in involving rankings between specific M and F constraints, not general classes thereof.[5] Which M and F constraints are crucially ranked depends on the structure of a given typology. For example, in Paka, m.LV (no long vowels) does not conflict with f.S (faithfulness to underlying stress value). While the constraints may be transitively ordered in some grammars, their relative ranking is not decisive in determining optima. Properties, and thus PERCs, precisely encode the rankings relevant to learning the grammars, rather than using unanalyzed blocks of M and F constraints.

However, having ERCs raises the prospect of committing the learner to rankings not directly evidenced by data and possibly inconsistent with the given target grammar. For this reason, PERCs are marked in the learner's grammar, and unlike ERCs obtained from winner-loser data pairs, can be retracted from a support during the first learning stage (PL) if inconsistent with the data.

4.1 *Algorithm* The algorithm begins by setting the learner's initial grammar (iG) to contain the PERCs form the PA. These are the M>F values of all the wide-scope MF Ps and are labeled as PERCs in the support. Learning then proceeds with PL, as in ODL, but differs in using RCD only rather than BCD. The learner extracts ranking information from the data based on taking the observed outputs as the inputs. If the data word is not correctly predicted to be optimal with the current ERC set, a WL pair is constructed and added to the support, using MRCD.

An inconsistent support in PL results when ranking requirements from data ERCs contradict those of the PERCs. The learner then identifies the PERCs in the URS of RCD and iterates through them, removing each and checking for consistency. If removal resolves the inconsistency, the learner updates the support by retracting the removed PERC and resuming PL.[6] Using the URS rather than the full support in the retraction step reduces the number of PERCs that are individually checked for consistency to those in the problematic subset. For Paka grammars, URSs contain a single PERC in each case, mainly because the PERCs involve different antagonist sets, though this will not always generalize to other systems.

The retraction method is only called during PL; after this stage, the learner commits to the remaining PERCs and in later stages they are treated like other ERCs in the support. During PL, the learner learns mainly F>M rankings. If the language contains a 'marked' form (one that violates a markedness constraint, such as a long vowel in the Paka system), the ERC from the WL pair will be inconsistent with a PERC enforcing neutralization, resulting in its retraction. The post-PL commitment to PERCs depends on the learner receiving a representative set of the data during PL, as is the case in (P)ODL. Inconsistency in later stages then indicates that feature values must be set, and PERCs can crucially be used in such inconsistency detection.

4.2 *Learning grammars* This section examines three distinct learning cases: i) the most restrictive grammar, L1; ii) a less restrictive grammar where PERCs are retracted, L20; and iii) the crucial case of L8, a paradigmatic subset of L7 that requires additional methods in ODL. For all, the learner begins with the same initial PERC set grammar, iG. In PA(Paka), the PERC-eligible Ps are P2: {m.ML,m.MR}.dom <>f.S, and P3: m.LV<>f.L, both set to their α value (7). These impose ranking conditions over different subsets of the constraint set. While the stratified hierarchy resulting from BCD on an empty support also places M constraints above Fs, the PERCs contain more specific and explicit ranking information.

[5] Smolensky's (1996) initial state H_0, like BCD, has all Ms > all Fs, though see his fn. 1.

[6] There is thus an ordering effect: the first inconsistency-causing PERC is retracted without checking if others might also resolve the inconsistency. The algorithm currently does not have another basis for choosing among multiple inconsistency-causing PERCs.

(7) Initial grammar (iG)

	m.ML	m.MR	m.NL	m.WSP	f.S	f.L
P2α	W	W			L	
P3α			W			L

4.2.1 *L1* L1 and its left-aligning counterpart L21 are the most restrictive grammars in the typology, neutralizing all inputs to a single output form, xX or Xx, respectively. These grammars are learned with PL alone in both ODL and PODL.

The learner begins with iG. The first—and only—data form used is xX. Assuming a faithful mapping, there are 2 possibly optima consistent with the PERCs: xX and Xx. The learner constructs a WL pair and adds the ERC shown in (8).[7]

(8) L1: 1st data word (and final G)

	m.ML	m.MR	m.NL	m.WSP	f.S	f.L
P2α	W	W			L	
P3α			W			L
xX:xX~Xx	L	W			W	

Fusion of this ERC and P2α yields LWeeLe (m.MR>m.ML & f.S). The learner's grammar is complete: no more errors occur, no underlying feature values require setting, and no PERCs are retracted. The PODL grammar contains 2 additional ERCs that are not in the final support produced by ODL, resulting in a more fully-specified grammar.

4.2.2 *L20* L20 is learned with PL, SF and CP. The language's phonotactic inventory consists of four forms: {Xx, xX, Hx, xH}: stress is lexical (faithful), with a left default position; long vowels shorten when unstressed. The full set of input-output mappings is shown in (9).

(9) L20 mappings

r1: x	r2: h	r3: X	r4: H	
Xx	Hx	Xx	Hx	s1: x
Xx	Hx	Xx	Hx	s2: h
xX	xX	Xx	Hx	s3: X
xH	xH	Xx	Hx	s4: H

With /Xx/ as the first word, 2 candidates tie for optimality, as in L1. The first WL pair is the opposite of that for L1, and in conjunction with P2α establishes the ranking m.ML>m.MR, f.S (10).

(10) L20: 1st PL word

	m.ML	m.MR	m.NL	m.WSP	f.S	f.L
P2α	W	W			L	
P3a			W			L
Xx:Xx~xX	W	L			W	

The next word is /xX/, with the opposite stress position. This word generates an error on the current support, where Xx is the predicted optimum. Adding the WL pair results in an inconsistent support when RCD is run (11). The joint ranking requirements of the 2 WL pairs can only be satisfied if f.S dominates both m.ML and m.MR, directly contradicting P2α. Contrastive stress is only possible if stress is lexical, (sometimes) faithful to underlying stress. The URS contains the three bolded ERCs. As only one of these is a retraction-eligible PERC, the learner removes it and checks for consistency. Without P2α, f.S can be placed in the next stratum, satisfying the remaining ERCs. The PERC is thus retracted.

[7] Inputs are stated in the form [input]: W~L.

(11) L20: 2nd PL word: Inconsistent RCD

	m.NL	m.WSP	m.ML	m.MR	f.S	f.L
P3α	W					L
P2α			**W**	**W**	**L**	
Xx:Xx~xX			**W**	**L**	**W**	
xX:xX~Xx			**L**	**W**	**W**	
Fus(2-4)			L	L	L	

The third word, /xH/, is mapped to xX under the current support. Again, the WL ERC results in inconsistency. The learner isolates the issue in the URS (bolded in (12)), and retracts the responsible PERC.

(12) L20: 3rd PL word: Inconsistent RCD

	m.WSP	f.S	m.ML	m.MR	m.NL	f.L
Xx:Xx~xX		W	W	L		
xX:xX~Xx		W	L	W		
P3α					**W**	**L**
xH:xH~xX					**L**	**W**

The last PL word, /Hx/, is the predicted winner under the current support. At the completion of PL, the PODL support is the same as that of ODL, as both PERCs were retracted. Subsequent learning stages of SF and CP proceed as in ODL, resulting in the final support in (13).

(13) L20: Final support

	m.WSP	f.S	m.ML	m.MR	f.L	m.NL
X.h:Xx~Xh	W				L	W
Xx:Xx~xX		W	W	L		
xX:xX~Xx		W	L	W		
xH:xH~xX					W	L
XX:Xx~xX			W	L		
XH:Xx~xH			W	L	L	W

4.2.3 *L8* L8 and its left-aligned image, L17 are the outliers among the 24 languages in the typology as the only grammars requiring FSF to be learned. The reason lies in the sub/superset relationship between L8 and another grammar in the typology, L7. Not only is L8's phonotactic inventory (3 forms) a subset of L7's (4 forms), but L8 is also a *paradigmatic subset* of L7: there are subparadigms within L7 that are surface-identical to L8 (outlined in bold in (14)). The L8 learner cannot rule out L7 on the basis of data alone, as it is also consistent with L7 (Tesar 2014:356).

(14) L8 and L7: Paradigmatic subset

	L8			L7				
Inventory	xX, xH, Hx			xX, xH, Hx, Xx				
Mappings	r1: x/r3: X	r2: h/r4: H		r1: x	r2: h	r3: X	r4: H	
	xX	Hx	s1: x s3: X	xX	Hx	Xx	Hx	s1: x
				xH	xH	Xx	Hx	s2: h
	xH	xH	s2: h s4:H	xX	xX	xX	Hx	s3: X
				xH	xH	xH	xH	s4: H

In ODL, following PL, the L8 grammar remains consistent with L7. BCD places f.S in the lowest stratum, but no ERC explicitly encodes this domination and thus there are no L's in the f.S column that can contribute to detecting inconsistency. Word testing indicates that additional features must be set, but there are multiple consistent value setting and no basis for choosing one over the other. For example, the word r1s1 has three unset features (only s1 is set as [-len]) (Tesar 2014:§8.3). Word evaluation indicates that

some feature(s) must be set, but there is not a unique setting that is consistent. Both values of [±len] for r1 are consistent, as shown in the final 2 rows of (15). The ODL learner uses FSF to choose the feature values, biasing toward the solution that requires setting fewer underlying features overall.

(15) Possible [len] values of r1

		m.WSP	f.L	m.NL	m.MR	m.ML	f.S
r2s2	hH:xH~hH	W	L	W			
r1s2	xH:xH~xX		W	L			
r2s1	Hx:Hx~xX		W	L	L	W	W
r1s1:r1[-len]	xX:xX~Xx				W	L	W
r1s1:r1[+len]	hX:xX~Hx		L	W	W	L	W

In PODL, the learner begins with iG. The occurrence of long vowels in /Hx/ results in the retraction of P3α, as in L20, leading to the grammar in (16).

(16) L8/7 grammar after L8 data

	m.WSP	f.L	m.MR	m.ML	m.NL	f.S
P2α			W	W		L
Hx:Hx~hX	W		L	W		W
xX:xX~Xx			W	L		W
xH:xH~xX		W			L	
Hx:Hx~xX		W	L	W	L	W

At this point, PL is complete for L8, but not for L7, where the additional data word /Xx/ adds a WL pair that is inconsistent with, and leads to the retraction of, P2α. In the L7 post-PL grammar, f.S dominates both m.ML and m.MR.

(17) L7 grammar post-PL

	m.WSP	f.S	f.L	m.NL	m.MR	m.ML
Hx:Hx~hX	W	W			L	W
xX:xX~Xx		W			W	L
xH:xH~Xx			W	L		
Hx:Hx~xX		W	W	L	L	W
Xx:Xx~xX		W			L	W

The retention of P2α in L8 makes it inconsistent with L7. The grammar is also inconsistent with [+len] value setting for the morpheme r1, shown in the bolded ERC in (18): there is no possible ranking of the constraints that satisfies all ERCs in the subset. P2α is crucial: without it, f.S prefers no losers and can be ranked in the highest stratum, leading to a consistent RCD result.

(18) L8: r1[+len] is inconsistent

	m.WSP	f.L	m.MR	m.ML	m.NL	f.S
P2α			W	W		L
Hx:Hx~hX	W		L	W		W
xX:xX~Xx			W	L		W
xH:xH~xX		W			L	
Hx:Hx~xX		W	L	W	L	W
hX:xX~Hx		**L**	**W**	**L**	**W**	**W**
fuse(1,3-6)		L	L	L	L	L

This inconsistency allows r1 to be set as [-len], and similarly for r3, which behaves exactly as r1 in L8. This setting is obtained in ODL through FSF, which does not add any WL ERCs nor is explicit about what

aspect of restrictiveness is at stake. The final L8 grammar under PODL thus differs from that in ODL in P2α, encoding the domination of f.S.

4.3 *Learning simulation results* Computer simulations for three versions of the (P)ODL algorithm were run on the Paka system, differing in the methods used, as shown in (19).[8] ODL1 is the version in Tesar (2014) and described above. A second version without BCD, ODL2, was also run, and required MMR to learn half of the languages, indicated in the final column below. PODL, which also does not use BCD, requires MMR for only 2 languages.

(19) Learning algorithms and results

	BCD	PERCs	FSF	MMR
a. ODL1	X		2	
b. ODL2				12
c. PODL		X		2

In PODL, the languages also differ in the degree of PERC retraction ((20); 'x' indicates that the PERCs is retained in the final grammars). L1 retracts no PERC, while L4 retracts all. PERC retention broadly tracks degree of restrictiveness.

(20) Retained PERCs

Language	P2α	P3α
L1	x	x
L3, 5, 8, 11	x	
L2		x
L4, 6, 7, 9, 10, 12		

5 Summary and discussion

PODL uses typological information from Property Theory to guide the learner towards salient rankings, particularly regarding restrictiveness. By defining the set of rankings that structure the typology, a property analysis isolates key conflicts that a learner can exploit.

PODL encodes a restrictiveness bias in ERCs in the learner's support in a way not used in previous ODL-based work that assumes a more cautious learner. To make PERCs feasible as a learning strategy, the learner must differentiate between two classes of ERCs during PL—PERCs and data ERCs—where only the former set is potentially defeasible. PERC retraction uses RCD, shown by Tesar (1995) to be highly computationally efficient.[9] The process of learning a less restrictive grammar in PODL thus involves ERC removal as well as ERC addition.

A key result of the algorithm is the elimination of BCD and reduction of the FSF and MMR methods, which have been shown to have limitations in more complex systems. BCD is not guaranteed to always find the most restrictive ranking, especially with certain kinds of faithfulness constraints (Prince & Tesar 2006). FSF was unsuccessful in Moyer's (2017) Paka extension, prompting the development of MMR. Limitations of MMR remain to be further examined. Additionally, the learned grammars in the PODL simulations are sometimes more complete than their ODL counterparts, due to containing ERC representations of rankings only implicit in BCD-generated hierarchies.

Not all Ps in the PA are used in PODL. For example, P1: m.ML<>m.MR, is an MM P and its values do not correlate with a difference in restrictiveness: neither default stress position is more 'restrictive' than the other.[10] Ps involving mixed classes, like P5 and P6, characterize more complex ranking relationships.

[8] Simulations used code written and maintained by Tesar (2010-19), modified for (b) and (c).

[9] The number of times PERC retraction is evoked during learning depends on the number of PERCs (wide-scope MF Ps) for that system.

[10] In a system with only MF Ps, there may be a unique 'most' restrictive grammar equivalent to the set of iG PERCs. In this case, no errors will be generated in PL and no ERCs added. If the system lacks MF Ps entirely, the PODL methods have no effect.

Depending on the values of other Ps that determine the dominant and subordinate members of the classes in P5 and P6, these values may or may not result in M>F rankings.

A potentially problematic kind of property for this method involves a class of Fs with the operator sub, as in {f1,f2}.sub. Ps with such antagonists do not occur in PA(Paka), but are frequent in PAs of other systems. The M>F value, where the F class is dominated, does not generate a single ERC (set) but a disjunction thereof (Alber & Prince in prep., DelBusso 2018). It is not possible to set a PERC in iG without knowing the identity of the subordinate member, an F<>F ranking. One possible approach is branching grammatical hypotheses (following Tesar 2004 and Akers 2012) for each disjunct, though this remains to be explored in greater detail.

Through properties, PODL taps into a level of theoretical structure not previously exploited in OT learning algorithms, and shows how such typological information can provide a way of enforcing restrictiveness and successfully learn subset grammars.

Appendix

Paka typology (right half): all inputs

Inputs	x.x	x.h	x.X	x.H	h.x	h.h	h.X	h.H	X.x	X.h	X.X	X.H	H.x	H.h	H.X	H.H
L1	x.X	x.X	x.X	x.X	x.X	x.X	x.X	x.X	x.X	x.X	x.X	x.X	x.X	x.X	x.X	x.X
L2	x.X	x.X	x.X	x.X	x.X	x.X	x.X	x.X	X.x	X.x	x.X	x.X	X.x	X.x	x.X	x.X
L3	x.X	x.H	x.X	x.H	x.X	x.H	x.X	x.H	x.X	x.H	x.X	x.H	x.X	x.H	x.X	x.H
L4	x.X	x.H	x.X	x.H	x.X	x.H	x.X	x.H	X.x	X.x	x.X	x.H	H.x	H.x	x.X	x.H
L5	x.X	x.H	x.X	x.H	h.X	h.H	h.X	h.H	x.X	x.H	x.X	x.H	h.X	h.H	h.X	h.H
L6	x.X	x.H	x.X	x.H	h.X	h.H	h.X	h.H	X.x	X.h	x.X	x.H	H.x	H.h	h.X	h.H
L7	x.X	x.H	x.X	x.H	H.x	x.H	x.X	x.H	X.x	X.x	x.X	x.H	H.x	H.x	H.x	x.H
L8	x.X	x.H	x.X	x.H	H.x	x.H	H.x	x.H	x.X	x.H	x.X	x.H	H.x	x.H	H.x	x.H
L9	x.X	x.H	x.X	x.H	H.x	x.H	H.x	x.H	X.x	x.H	x.X	x.H	H.x	H.x	H.x	x.H
L10	x.X	x.H	x.X	x.H	H.x	h.H	h.X	h.H	X.x	X.h	x.X	x.H	H.x	H.h	H.x	h.H
L11	x.X	x.H	x.X	x.H	H.x	h.H	H.x	h.H	x.X	x.H	x.X	x.H	H.x	h.H	H.x	h.H
L12	x.X	x.H	x.X	x.H	H.x	h.H	H.x	h.H	X.x	x.H	x.X	x.H	H.x	H.h	H.x	h.H

References

ROA: Rutgers Optimality Archive (roa.rutgers.edu)

Alber, Birgit DelBusso, Natalie & Prince, Alan. 2016. From intensional properties to Universal Support. *Language: Phonological Analysis* 92(2): e88-116. ROA 1235.

Alber, Birgit & Prince, Alan. In prep. *Typologies*. Ms, University of Verona & Rutgers University.

Akers, C. 2012. *Commitment-based learning of hidden linguistic structures*. PhD dissertation, Rutgers University.

Brasoveanu, Adrian & Prince, Alan. 2011. Ranking and necessity: The Fusional Reduction algorithm. *Natural Language & Linguistic Theory* 29.1: 3-70.

DelBusso, Natalie. 2018. *Typological structure and properties of Property Theory*. PhD diss., Rutgers University.

Merchant, Nazarré & Prince, Alan. To appear. *The mother of all tableaux*. Advances in Optimality Theory. Sheffield & Bristol, CT: Equinox. ROA 1285.

Moyer, Morgan. 2017. Output-driven learning, restrictiveness and in inductive leap: Learning stressless prosodic systems from distributional evidence. Qualifying paper, Rutgers University.

Moyer, Morgan & Tesar, Bruce. 2019. Enforcing restrictiveness through ranking induction in the Output-Driven Learner. 93rd Annual Meeting of the Linguistic Society of America, New York, Jan. 3-6.

Prince, Alan. 2002. Entailed Ranking Arguments. Tech. report, Rutgers University. ROA 500.

Prince, Alan. 2017. Representing OT grammars. Tech. report, Rutgers University. ROA 1309.

Prince, Alan. & Tesar, Bruce. 2006. Learning phonotactic distributions. Tech. report, Rutgers University. ROA 353.

Smolensky, Paul. 1996. The initial state and 'Richness of the Base' in Optimality Theory. Tech. report JHU-CogSci-96-4. ROA 154.

Tesar, Bruce. 1995. *Computational Optimality Theory*. PhD dissertation, University of Colorado, Boulder.

Tesar, Bruce. 1997. Multi-Recursive Constraint Demotion. Ms., Rutgers University. ROA 197.

Tesar, Bruce. 2004. Using inconsistency detection to overcome structural ambiguity in language learning. *Linguistic Inquiry* 35.2: 219-253.

Tesar, Bruce. 2014. *Output-Driven phonology: Theory and learning*. Cambridge: Cambridge University Press.