Optimality Theoretic Rankings with Tied Constraints: Slavic Relatives, Resumptive Pronouns and Learnability

Kevin Broihier

Department of Brain and Cognitive Sciences

MIT

Optimality Theoretic Rankings with Tied Constraints:

Slavic Relatives, Resumptive Pronouns and Learnability

Introduction

Using the formal machinery of Optimality Theory (OT) (see, for example, Prince & Smolensky, 1993), Pesetsky (1993) analyzes and explains the conditions under which syntactic elements that interact with the CP systems of French and English receive overt phonological realization[1]. This paper extends Pesetsky's results in two ways.

First, I argue that the constraint set and the augmented OT apparatus that Pesetsky develops combine, almost directly, to yield an account of an interesting interaction in Polish relative clauses between the morphologically invariant element 'co' and a set of elements that I'll analyze as resumptive pronouns. Given this analysis of Polish, a number of properties of Russian relative clauses also fall out of the typology inherent in Pesetsky's system.

Second, I address a gap in the theory of the learnability of OT systems that Pesetsky's extension of the standard OT framework opens. From his analysis of English and French syntax, Pesetsky provides arguments for relaxing the requirement that all constraints be ordered in a strict dominance hierarchy. He also provides a precise procedure for computing the optimal members of a candidate set when grammars take advantage of this new possibility by "tying" constraints[2]. The only explicitly proposed general algorithm for acquiring language specific orderings of universal OT constraints (Tesar & Smolensky, 1993) fails to acquire any target language that

---

1 Unless otherwise noted, all further references to Pesetsky will be to Pesetsky (1993).

2 It's an open question whether this move will prove necessary in other domains. There are reasons to doubt that there will be a real abundance of ties.

requires Pesetsky's, or, indeed, any other notion of tied constraints[3,4]. The need for a learning algorithm that can demonstrably handle tied constraints becomes even more pressing in light of the analysis of Polish relatives presented in the first half of the paper; this analysis goes Pesetsky one better (or worse!) by relying not only on a two-way, but, crucially, on a three-way tie among constraints. Demonstrating and diagnosing the difficulties that tied constraints cause for Tesar and Smolensky's version of the constraint demotion (CD) algorithm will take up the initial portion of the second half of the paper. The mood of most of this second half, however, is generally upbeat, though tentative. I explore two modifications of the CD algorithm that succeed in acquiring the correct constraint rankings for the system's grammar of English and consider their applicability to Polish. The modifications that I consider promise to extend generally to a class of languages that we might hope to find the class of natural languages contained within.

Part 1: The Realization of Resumptive Pronouns in Slavic Relative Clauses

Optimality Theory

To date, the analytic tools of OT have primarily been deployed in the domain of phonology. Nothing about OT's basic conception of the

---

3 It's possible to produce degenerate cases where the same array of data can be analyzed, in terms of a set of constraints, both with and without ties. The Tesar and Smolensky (1993) algorithm, reasonably, takes coverage of the data as its criterion of success. Any pattern of data which doesn't require a tie for its analysis will drive the algorithm to acquire a strictly ranked ordering.

4 Unless otherwise noted, all further references to Tesar and Smolensky will be to Tesar and Smolensky (1993).

formulation of linguistic knowledge, however, limits its application to this domain and a number of researchers, including Grimshaw (1993), Legendre, Raymond, & Smolensky (1993), and, of course, Pesetsky, have begun to develop OT accounts of syntactic phenomena that have more typically been treated in (or perhaps noticed but left largely unanalyzed in) a Principles and Parameters framework. I won't review the motivations for the OT approach here (see further, Prince & Smolensky (1993)), but I will briefly discuss its mechanics. The analyses I'll present below rely heavily on an understanding of the basics of OT's machinery.

OT accounts view the structures generated by natural languages as the survivors of competitions among the members of sets of candidate forms. These candidate sets are constructed by a cross-linguistically universal function called GEN, for "generate", using the basic representational formats provided by Universal Grammar (UG). GEN initiates competition among forms by taking an input and generating the appropriate candidate set. For example, in the domain of metrical phonology, GEN might take a string of segmental material from the lexicon and return a set of possible candidate parses of this material into syllables, mora, feet and prosodic words. (Of course, languages may use different lexicons to feed GEN, but, on the standard view, GEN's operations remain universally fixed.) In Pesetsky's system, as will be discussed below, the inputs to GEN are syntactic structures of familiar sorts; Pesetsky considers two possible input representations for GEN—S-structures and Minimalist-style LFs (Chomsky, 1992). Given these syntactic inputs, GEN produces a variety of candidate pronunciations of the input that differ with respect to which of the input's constituent parts receive overt phonetic realization. The competition between candidate forms is mediated by a set of universal well-formedness or "well-pronouncedness" constraints.

Unlike principles of the more canonical type, Pesetsky's well-formedness constraints can, like all canonical OT constraints, be violated. The forms that occur in any particular natural language are not forced to exhaustively satisfy the entire constraint set. This is a good thing, because in this case, as in the typical OT case, the constraints conflict massively and complete satisfaction would be impossible. On the OT view, then, failure to satisfy a constraint does not necessarily result in ungrammaticality. Rather, it becomes necessary to select a candidate that satisfies the constraints as best as it can, or *optimally* . For selection of the optimal candidate to take place, of course, we require some means of measuring which candidate does the best job of satisfying a constraint set. Logically, many different constraint accounting calculi are possible (see for example the essays collected in Goldsmith (1993) for some alternative proposals). Prince and Smolensky (1993), however, propose that constraints are strictly ranked in order of importance and Pesetsky starts with this standard OT assumption. Given a set of ranked constraints, selection of the optimal candidate(s) from a candidate set happens like this[5]:

> To best satisfy a system of well-formedness constraints means the following. Except for ties, the candidate that passes the highest-ranked constraint is the output form. A tie occurs either when more than one candidate passes the highest-ranked constraint or when all candidates fail the highest-ranked constraint ... In cases of ties, all surviving candidates are tested recursively against the next level of the hierarchy. Once a victor emerges, the remaining lower-ranked constraints are irrelevant; whether the sole surviving candidate obeys them or not does not affect its grammaticality. Likewise the evaluation of failed constraints by lower-ranked constraints is also irrelevant; no

---

[5] It's worth pointing out that a tie between candidates is very different than a tie among constraints—the innovation that Pesetsky introduces.

inferences about degree of deviation from grammaticality can be drawn from further inspection of failed candidates.

On the OT view, grammars differ precisely in the ranking of these constraints, which, by hypothesis, are universal. The learner’s task is to use input data from the target language to correctly rank these constraints.

Pesetsky (1993): Initial Framework

Pesetsky uses the OT framework's conceptual tool kit to explain both intra-linguistic and cross-linguistic patterns in the overt realization of various elements that enter into the CP system. The idea that the distribution of these items is sensitive to the special phonetic status of the edges of syntactic categories such as CP guides much of the work. I'll be discussing two versions of the system. In the first version, the candidate sets are generated from S-structure forms. In the second version, the candidate sets are generated from LF forms that respect some version of the Copy Theory of Movement (Chomsky, 1992). This organization reflects the historical development of the system, and I believe retaining it in my presentation will be useful. The first version of the system has been worked out in most detail and is somewhat less complicated. The constraint set of the second version of the system is essentially an extension of the first. It incorporates the constraints of the first version and introduces several constraints of its own to regulate the larger candidate sets generated by the move to LF forms. My analysis of Polish and Russian takes the second version as its starting point.

The candidate sets in the first version of the system consist of all possible “pronunciations” of linearly ordered, phonetically specified, S-structure forms. An individual S-structure form is input to GEN and GEN returns the set of candidate pronunciations, which are generated by producing all possible labellings of a class of S-structure constituents with the feature [+/- silent]. As the system has developed, certain lexical phrases and certain heads qualify as possible candidates for [+silent] marking. In this version of the system, there are two important sources of cross-linguistic variation. The mechanisms that construct S-structures are retained from work rooted in a more familiar Principles and Parameters framework, while the mechanisms that generate and decide among various pronunciations of an S-structure are cast in terms of an OT system of constraints fed by GEN.

Since elements marked [+silent] eventually fail to receive a phonetic realization, phonological material will only be pronounced when it is not labeled [+silent] in the optimal candidate. In effect, the syntactic elements in a sentence are forced to struggle for their phonetic survival. A set of violable constraints with an intricate interaction determines the outcome of this struggle. Some of these constraints simply state either a preference for, or an objection to, the [+ silent] pronunciation of particular constituents. If the system only had this type of constraint, it would be fairly uninteresting. Linguistic variation could arise only if both *[+silent] and *[-silent] constraints existed for a given syntactic type; rankings of the relevant *[+silent] and *[-silent] constraints would then directly determine pronunciation regardless of the ranking of other constraints. Things, of course, turn out to be more interesting. Added to the mix is a set of constraints that favor overt phonetic realization for particular units that stand in adjacency relations to particular syntactic edges. The addition of this type of constraint transforms the system, in part, into a competition between different components of a sentence because being pronounced in a privileged edge position often requires a sentence component to make sure other elements blocking adjacency to the edge delete[6].

Pesetsky's initial constraint set is listed in (1)[7, 8].

(1)

(a) RCV: Deleted (i.e. +silent) material must be recoverable
(b) LEC: A complementizer must be pronounced at the left edge of CP
(c) TEL: A function morpheme must be unpronounced
(d) ECP: Elements in CP must be pronounced when CP is not head-governed
(e) LE(F): A finite verb must be pronounced at the left edge of CP
(f) LE(to): Infinitival 'to' must be pronounced at the left edge of CP

To briefly exemplify the interaction between the trio of RCV, LEC and TEL that will concern us here consider the following OT tableaux for French finite relative clauses. (The other constraints are not relevant here.) On Pesetsky's analysis, French obeys the following constraint ranking: RCV >> LEC >> TEL. I'll be adopting several typographical conventions throughout: (1) the symbol ">>" will be used to indicate ordering in the language's dominance hierarchy; (2) violations of a constraint are represented in the appropriate column with the mark, "*"; (3) violations which knock a candidate out of consideration are

6 It is something of a misnomer to speak of a morpheme "making sure" that something happens since the system operates by selecting among a static set of structures. I will, however, continue to do so where this intuitive way of speaking does not obscure my point.

7 My enumeration of the constraints follows the treatment that Pesetsky provided in his course Spring Term at MIT. Pesetsky (p.c) has suggested that LE(F) may not work out as part of the system.

8 The output of the system is, in some cases, subject to filtering; it is possible for a derivation to be selected by the OT system and then "crash". If every optimal output from a candidate set crashes structural ineffability results. Some of these crashes can be caused by the failure to mark [+silent] elements which have no interpretable phonetics. These devices will not be used in the present work.

additionally marked with a "!"; (4) shading serves as a reminder that the constraints ranked lower than these "fatal" constraints are irrelevant to the outcome of the optimality computation; (5) the appearance of a "=>" before the first constraint column of a row indicates that the candidate is an optimal one; (6) the "*" actually does double duty and continues to serve its traditional role by indicating structures that are ungrammatical in the language under analysis; (7) deleted material, that is to say those elements marked [+ silent], are indicated by "Ø".

Tableau (1):

	RCV >>	LEC>>	TEL
1*l'homme [$_{CP}$ qui que je connais		*!	*
2*l'homme [$_{CP}$ qui Ø je connais		*!	
3 l'homme [$_{CP}$ Ø que je connais =>			*
4*l'homme [$_{CP}$ Ø Ø je connais		*!	

the man (who) (that) I know

Tableau (1) illustrates the competition between several variant "pronunciations" of a French finite relative clause with a relativized object NP. On the assumption that the relativized NP, 'qui', is able to delete recoverably because of its relation to the head of the relative clause, RCV has no effect on the analysis[9]. The absence of any marks in RCV's column reflects this. The task of selecting the optimal candidate, therefore, falls on the more lowly ranked constraints. LEC, the next in line in the strict dominance hierarchy, directly determines the output by eliminating candidates 1-1, 1-2 and 1-4. Candidates 1-1 and 1-2 overtly spell out the NP in a position where it

[9] Exhaustively spelling out the content of this notion is not a project that I intend to undertake here. I will, however, empirically constrain this notion by distinguishing by identifying differences in recoverability as the source of differences in the behavior of different classes of NPs in Polish and Russian.

would block adjacency of the complementizer, 'que' and the left edge of CP. Material that has been marked [+silent] is invisible in the computation of adjacency, so candidate 1-3 incurs no mark from LEC. Candidate 1-4 deletes C and, by doing so, dooms any hope of satisfying LEC's requirement that the complementizer 'que' receive overt realization at the left edge of CP. Material that has been marked [+silent] can not satisfy LEC's adjacency requirement.

Tableau (2):

	RCV >>	LEC>>	TEL
1*l'homme [$_{CP}$ avec qui que j'ai dansé		*	*!
2 l'homme [$_{CP}$ avec qui Ø j'ai dansé =>		*	
3*l'homme [$_{CP}$ Ø que j'ai dansé	*!		
4*l'homme [$_{CP}$ Ø Ø j'ai dansé	*!		

the man (with whom) (that) I danced

Tableau (2) illustrates a dramatic shift in the behavior of the system when the relativized phrase is a PP. By hypothesis, PPs do not recoverably delete. RCV, therefore, immediately eliminates 2-3 and 2-4, the two variants where the PP gets suppressed; later satisfaction of LEC can't compensate for this more highly ranked violation. LEC does not decide between the remaining candidates since both violate it. It falls to TEL to eliminate the "doubly filled COMP" violation in 2-1 and leave the pronunciation where the PP surfaces and C deletes to emerge[10].

As shown in Tableau (3), the same constraint ranking directly predicts the behavior of the complementizer in French embedded declaratives. With

[10] Of course the Doubly Filled COMP filter plays no real role in the system.

no material in [SPEC, CP], C is already at the left edge of CP and the form that overtly pronounces the complementizer emerges as the optimal candidate.

Tableau (3):

	RCV >>	LEC>>	TEL
1 Je crois [CP que Pierre a faim =>			*
2*Je crois [CP Ø Pierre a faim		*!	

I believe (that) Pierre has hunger

So far, it has been possible to apply the constraints in entirely conventional OT style. The optional realization of C in English embedded declaratives, shown in (2), directly raises problems for this approach.

(2a) I believe that Peter is hungry.

(2b) I believe Peter is hungry.

Since the constraints that operated in the analysis of French are supposed to apply universally, free choice between forms that incur different sets of marks should be impossible; at some point in the comparison of violations, one candidate will be eliminated. Empirically, however, both candidates emerge from the English "translation" of the candidate set in Tableau (3). Pesetsky solves this problem by proposing that English grammar ties the constraints LEC and TEL. The notion of tie, however, has no interpretation in the system so far. Pesetsky considers three possible interpretations of this idea. Under all three candidate interpretations, the blocks of tied constraints effectively function as a single “meta”-constraint whose mark-assigning procedure can be built up compositionally from its constituent filters. First, under a “branching” or “logical or” notion of tied constraints, satisfaction of any one of the tied constraints suffices for satisfaction of the entire tie block. Second, under a "pooled violation" notion of ties, the marks from the tied constraints are combined in one column and evaluation proceeds, otherwise, as normal. Under a "reordering" notion of ties, the candidates are run through each possible reordering of the tied constraints in parallel; the candidates that successfully emerge under any of these reorderings incur no mark from the meta-constraint; all others do incur a mark. Although all three notions will work in this case, Pesetsky provides arguments for the third "reordering" notion and the analyses in the present paper will strengthen those arguments. To see how this solves the problem for English embedded declaratives, note that the form where C is silent is the optimal candidate under the ordering TEL >> LEC. As an aside, it is worth noting that the choice that the data force on Pesetsky is in many senses the most restrictive extension of the standard OT system. The output from a system which forms "reordering" ties will be a subset of the union of the outputs from a set of strictly ranked systems. Only forms that occur somewhere in the typology of strictly ranked grammars can occur in a "reordering" tied grammar. This is not the case for either of the other notions[11].

A consideration of the predictions that follow for simple relative clauses in English counters objections that the introduction of ties is an *ad hoc* solution to the problem raised by optional deletion of C in embedded declaratives. The tie between LEC and TEL almost directly predicts the greater

---

[11] For example, consider a dummy filter which assigns no marks to any form. On the branching interpretation of ties, tying this dummy filter with LEC and TEL would allow for the reemergence of Doubly Filled Comp violations. This would not be the case under the reordering notion.

range of variation possible when NPs are relativized in English, as compared to French. This range is illustrated in (3)[12].

(3a) *the man who that I know

(3b) the man who I know

(3c) the man that I know

(3d) the man I know

Consideration of the two possible reorderings of the two-way tie of constraints indicates how this pattern follows directly from the introduction of ties.

Tableau (4a):

	RCV >>	LEC>>	TEL
1*the man [CP who that I know		*!	*
2 the man [CP who Ø I know		*!	
3 the man [CP Ø that I know =>			*
4 the man [CP Ø Ø I know		*!	

Tableau (4b):

	RCV >>	TEL>>	LEC
1*the man [CP who that I know		*!	*
2 the man [CP who Ø I know =>			*
3 the man [CP Ø that I know		*!	
4 the man [CP Ø Ø I know =>			*

As in French, the "complementizer only" form wins out when LEC dominates TEL. When TEL dominates LEC, however, the complementizer must be suppressed. The filters don't distinguish between cases that differ

12 The proposal actually overgenerates slightly. When a subject NP is relativized the otherwise acceptable "all null" pronunciation is blocked, as in (a):

*(a) I saw the man [CP knows me]

See Pesetsky for an effort to resolve this problem. I'll also be discussing this late in the section.

only in their pronunciation of recoverable material. Therefore, two possibilities emerge.

<u>The Extension of Pesetsky (1993) Yields a Taxonomy of Resumptive Pronouns in Non-Island Positions</u>

The first version of the system receives the most attention in Pesetsky's manuscript. Pesetsky, however, concludes with a consideration of a possible extension of the system to cover a wider cross-linguistic array of movement phenomena including cross-linguistic variation in wh-interrogative, relative, A-movement, scrambling and island violation constructions. This extension is what I'm calling the second version of the system; one of its goals is to reduce the work done by non-OT components of the system by subsuming "movement" phenomena which the first version relegated to the more standard Principles and Parameters component. The treatment of Polish and Russian relative clauses presented here develops naturally from this second version. The extended system replaces the S-structure style inputs to GEN with something like fully pronounced, linearly ordered, Minimalist logical forms as the input to GEN's marking procedure. This move expands the candidate sets involved in any computation of the optimal candidate because now variants with and without pronounced "traces" compete against each other. Moreover, GEN also now distinguishes three gradient levels of pronunciation to lexical phrase [+ silent], [reduced] and [-silent]; [reduced] phrases are realized as proforms. From the point of view of constraints that block pronunciation, [reduced] pronunciation is a less serious violation than [-silent] pronunciation. To regulate this expanded competition, Pesetsky adds three filters that place additional constraints on the set of pronounceable chain positions. Following out the consequences that follow from the addition of one of these filters—Minimize Trace—will lead us to an account of relative clauses in Polish and Russian[13]. The three "chain-regulating" filters are:

(4a) Low Resolution[14]:
Pronounce an element in the lowest position where its strong features have all been checked off.

(4b) Minimize Trace:
Provide traces with minimal pronunciation—[+ silent] where possible, [+ pronominal] elsewhere.

(4c) Satisfy Islands:
Disallow [+silent] in certain configurations[15].

In this new system, recoverability is now crucially understood as a property of chains; either [reduced] or [-silent] pronunciation of any link in the chain is enough to ensure that RCV is satisfied by every link in the chain.

The adoption of Minimize Trace fairly directly predicts the existence of languages where moved prepositional phrases are realized, either obligatory or optionally, as some sort of resumptive proform. Resumption will, in some

---

13 The OT strategy of maintaining that all languages deploy the set of constraints obligates me to also include Low Resolution and Satisfy Islands in my analysis of Polish and Russian, provided that I believe that they do work elsewhere. However, these constraints have no visible effect in the paradigms I consider.

14 As suggested above, the constraint set is a work in progress. Pesetsky (p.c.) has moved towards the conclusion that the notion of feature strength should not be imported into the constraint system from the Minimalist framework. This move would be consistent with the stated goal of providing a reanalysis of movement phenomena.

15 Clearly, more remains to be said here.

cases, be mandatory even when PPs have not moved from island positions; the remainder of this section derives these predictions[16].

Under the OT framework's default assumptions, providing a system of constraints and a set of candidate sets for these constraints to operate over leads directly to a typology of possible languages. The method for generating this typology is simple: Generate all possible orderings of the constraints and, in turn, apply them to each of the candidate sets[17]. Each set of sets of optimal candidates that successfully navigate through a particular constraint ordering represents the possible outputs in the language generated by that ordering. In many cases, several different constraint rankings can produce the same output.

Adding ties expands the set of constraint orderings that must be considered but the basic procedure for typology generation does not change

---

16 An important improvement in the treatment of word order becomes possible at this point. The filters in Pesetsky's initial system, as it stands, do not take a completely universal candidate set as their input. Word order, is understood as fixed by a parametrized system which provides the [+silent] marking procedure with its inputs. Non-universality of the candidate sets feeding into the OT system would intensify the need to cash in the assumption that the learner can compute information about the competitors of a form attested in the target language. The complexity introduced by the interaction of these two components is typically avoided in phonology by making candidate sets cross-lingustically invariant. If different word orders compete against each other in the OT system, it becomes possible to suppose that candidate sets are universal in this case as well. Cross-linguistic differences in word order are, on this view, analyzed as a reflex of the constraints regulating chain spell out operating on the output of a universally specified GEN. This GEN would take an array of lexical items as input.

17 Actually, when generating typologies, the candidate sets considered can be more abstract than the candidate set considered in the generation of an actual word or sentence.

dramatically. It's simply necessary to keep track of all the possible ways of tying sets of filters. The new procedure is as follows: Generate all possible partitions of the constraints with the understanding that constraints grouped together within a partition get interpreted as tied. Generate all possible orderings of the constraints implied by the ties and, in turn, apply them to each of the candidate sets[18]. Each set of sets of optimal candidates that result from considering all tie-compatible reorderings of a particular partition of the constraints represents the possible outputs in the language generated by that ordering.

So, what typology of simple relative clauses emerges from the taxonomy implicit in Pesetsky's system? I will focus on a partial answer to this question and will limit consideration of the full typology in two ways. First, I'll only consider the ordering of the constraints: MIN, LEC, RCV and TEL[19]? Second, RCV will be the highest ranked constraint in all the systems that I consider. These limitations require discussion and justification. The justifications for focusing on these four constraints and temporarily excluding others from consideration take one of two forms. Either the constraint can't ever differentiate relative clause candidates in the tableaux I'll consider, or placing the constraint below LEC will mask its effects. ECP and Satisfy Islands provide examples of constraints of the first type. Neither constraint can, as the system stands, play a role in determining the types of simple relative clauses

---

18 Applying a set of tied constraints, as described above, itself involves considering all possible orders of the constraints in the tie block.

19 The computation of possible relative clauses was performed by a set of procedures written in Common LISP.

involving local wh-movement[20]. The neglect of these constraints, of course, can only be temporary. Once relative clause data from a language forces certain rankings of MIN, LEC, RCV and TEL, it becomes possible, and important, to consider what happens in different constructions where ECP and Satisfy Islands enter the computation. Temporary neglect of LE(F), on the other hand, genuinely restricts the scope of inquiry even within the domain of simple relative clauses. It is, however, possible to place this constraint low enough that it has no effect on relative clauses. For example, ranking LEC higher than LE(F) voids LE(F) of any effect; if LEC is satisfied, LE(F) can't be since only one element can occur at the left edge of CP[21]. However, it's entirely possible to generate additional relative clause types by upgrading the constraint's ranking. I will not be exploring these cases here, although doing so would obviously provide a further important test of the system's predictions. The decision to focus on the "high RCV" subset is based on an empirical hunch that this constraint may be universally undominated and a desire not to clutter discussion in the text. For sake of completeness, though I present the full taxonomy, including grammars that don't rank RCV first, in an appendix.

The displays in (5) through (9), then, illustrate the relative clause typology generated by this limited system. I discuss the typological possibilities in turn. I've introduced a "bracketing and equation" notation to indicate tied constraints.

---

20 Pesetsky assumes that the CP in a simple relative is head-governed and, therefore, does not violate ECP. ECP does play a role in the system for stacked relatives, as will be seen below.

21 The question becomes more vexed if constraints like TEL are eventually subdivided into constraints like TEL(C) and TEL(F). I will for now, simply note and then ignore this possibility.

(5) No Complementizers, No Resumptives

<u>Possible Pronunciations</u>:

NP_i C $trace_i$	=>	NP	Ø	Ø
		Ø	Ø	Ø
PP_i C $trace_i$	=>	PP	Ø	Ø

<u>Rankings</u>:

(a) RCV >> TEL >> LEC >> MIN

(b) RCV >> MIN >> TEL >> LEC

(c) RCV >> TEL >> MIN >> LEC

(d) RCV >> (TEL = MIN) >> LEC

(e) RCV >> TEL >> (MIN = LEC)

Figure (5) illustrates the effects of ranking TEL >> LEC. Remember that the engine that drives the competition *between* morphemes is the system's set of edge conditions on morphological items. In certain configurations, a morpheme's pronunciation at the relevant syntactic edge—CP in the explicitly developed constraints—necessarily occurs at the expense of other phonological material that intervenes. If a morpheme, like C, is subject to a TEL constraint that disfavors its overt realization, then placing this constraint above any edge constraints governing the morpheme, like LEC in the case of C, will make it impossible (without, of course, the introduction of other constraints) for the "telegraphable" morpheme to surface. This is exactly what happens given the orderings in (5a) through (5e). Alternation between the two possibilities for NP realization is allowed because no filter is directly sensitive to the NP's presence.

(6) "French"

Possible Pronunciations:

NP_i C $trace_i$	=>	Ø	C	Ø
PP_i C $trace_i$	=>	PP	Ø	Ø

Rankings:

(a) RCV >> MIN >> LEC >> TEL

(b) RCV >> (MIN = LEC) >> TEL

Provided that MIN, the system's new constraint, remains high, the new members of the candidate set that spell out the chain as a proform in the base position of the chain have no chance of surfacing because the mechanism for candidate set creation insures that a candidate with a resumptive pronunciation will always be in competition with an otherwise identical candidate with no resumptive. With MIN sufficiently high, the new resumptive candidates can never surface and LEC >> TEL generates the now familiar paradigm from French, summarized in (6).

(7) "English"[22]

Possible Pronunciations:

NP_i C $trace_i$	=>	NP	Ø	Ø
		Ø	C	Ø
		Ø	Ø	Ø
PP_i C $trace_i$	=>	PP	Ø	Ø

Rankings:

(a) RCV >> (LEC = TEL) >> MIN

(b) RCV >> MIN >> (LEC = TEL)

22 When a subject NP is relativized, English does not allow for null pronunciation of both the complementizer as this analysis will predict. Discussion of this problem will be deferred until later, when it surfaces again in discussion of the Slavic examples.

As discussed above in the consideration of English, tying LEC and TEL strips LEC of its power to force material to delete from [SPEC, CP]. Given that it is possible to pronounce the higher link of a wh- chain that occupies this position, it becomes impossible to pronounce the dispreferred resumptive alternative. Because of this, MIN's position in the constraint hierarchy is irrelevant, as reflected in (7).

(8) Obligatory Resumption for Unrecoverable Elements

Possible Pronunciations:

NP_i C $trace_i$	=>	Ø	C	Ø
PP_i C $trace_i$	=>	Ø	C	"trace"

Rankings:

(a) RCV >> LEC >> TEL >> MIN

(b) RCV >> LEC >> MIN >> TEL

(c) RCV >> LEC >> (MIN = TEL)

As seen in (8), degrading the status of MIN, when LEC and TEL are untied, makes it possible for forms with resumptive proforms to emerge as optimal members of a candidate set. Introducing the resumptive structures into the candidate set provides relative clauses with an alternative way of satisfying RCV without pronouncing material in [SPEC, CP]. Opening this new option deprives top-ranked RCV of the status it initially enjoyed in the system as an absolute barrier to the deletion of unrecoverable material in [SPEC, CP]. Recoverable material in [SPEC, CP] still disappears under the leftward pressure from the complementizer. [SPEC, CP], however, is no longer an exclusive venue for the pronunciation of unrecoverable wh-material. Since RCV doesn't discriminate between chain positions it is possible, and therefore necessary, to take the resumptive option.

(9) Resumption Alternates with Complementizer Only for Unrecoverable Elements

Possible Pronunciations:

NP_i C $trace_i$	=>	NP	Ø	Ø
		Ø	C	Ø
		Ø	Ø	Ø
PP_i C $trace_i$	=>	PP	Ø	Ø
		Ø	C	"trace"

Rankings:

(a) RCV >> (LEC = MIN = TEL)

In previous cases, several different orders would do an equally good job of handling the relative clause data and, in fact, only the English pattern required any tied constraints. The final pattern that the subsystem under consideration generates also requires a tie. The ranking indicated in (9) produces quite an array of possible pronunciations for simple relative clauses. This full range of data can only occur when there is a three-way tie between LEC, MIN and TEL.

Since, as I've laid it out above, the method of interpreting the consequences of n tied constraints requires the evaluation of $n!$ tableaux, where n is the number of tied constraints, it's more difficult to directly visualize how the proposed tie generates this result. I will defer presenting these six relative clause tableaux until the next section because my analysis of Polish relative clauses directly makes use of this three-way tie.

Resumptive proforms are often understood as somewhat deviant efforts to repair violations caused by extractions that violate linguistic principles. From this point of view the occurrence of what appear to be resumptive elements in cases of very local movements to [SPEC, CP] would be somewhat problematic since such movement does not seem to incur such violations. In the current analysis, however, resumptive proforms don't necessarily receive a uniform treatment as a diagnostic of island violations. Instead, they can arise in a variety of circumstances when pronunciation of a higher link in the chain is non-optimal.

Polish Relative Clauses

The taxonomic analysis of Pesetsky's system has led to the prediction of the existence of languages that either obligatorily or optionally pronounce a resumptive form in extracted position when PPs relativize. A (hopefully) short stretch of the imagination, requiring some willingness to negotiate over the placement of the border line between recoverable and non-recoverable elements, allows an analysis of an interesting set of structures in Polish that include both what I'll call short resumptive relative clauses and more mundane relative clauses realized by pronunciation of a case-marked wh-word and omission of the complementizer. The stretch involves considering the possibility that certain case morphology also qualifies as unrecoverable. On the assumption that English lacks such unrecoverable case morphology Pesetsky's analysis remains unchanged; English NPs are freely deletable in cases of relativization. In languages where NPs can bear unrecoverable morphology, however, those NPs should behave as PPs are predicted to[23]. I'll argue that this is the case in Polish. Examples discussed in this section comes from either Fisiak, Lipinksa-Grzegorek and Zabrocki (1978) or Golab and Friedman (1972).

---

23 Fox (manuscript) argues that resumptive pronouns arise in cases where case is marked in PPs. Extraction of NP from within a PP results in an island variation, thereby potentially placing even short resumptives under the purview of Satisfy Islands.

The form that Polish relative clauses take displays an interesting sensitivity to the case-marking of the relativized NP. In all cases, it is possible to produce a relative where a "wh-form", appropriately case-marked, surfaces in [SPEC, CP] without an overt complementizer or resumptive proform within the relative. This form is, by now, familiar from the discussion of English and French. Polish also, however, allows relatives with an invariant form 'co' occurring in the same position that the case-marked wh-phrase appears in for the first class of relatives. The wh-phrase and 'co' are in complementary distribution. Its positioning and morphological invariance lead me to analyze 'co' as a complementizer in these constructions[24, 25]. These same phenomena, of course, call out for an analysis in Pesetsky's system, which is geared to handle just such complementarities. It is in these cases involving 'co' that case-sensitivity emerges. When the subject NP is relativized, there is no spell-out of the relativized NP in the lower clause. This is exemplified in (10), taken from Fisiak, Lipinksa-Grzegorek and Zabrocki (1978) .

---

24 Morphological invariance of the complementizer is, of course, not necessary (see for example Rizzi (1989) for discussion of agreement marking on C).

25 'Co' also appears in interrogative constructions where it is typically glossed as 'what' as in:

(a) Co zjadl Janek?
what ate John?
"What did John eat?"

No pronominal occurs with these uses. This observation suggests the possibility of arguing that recoverability is simplified in interrogative environments, making it easier for invariant 'co' to succeed in its push to the left. Pursuing this possibility will require careful formulation of the notion of recoverability.

(10a) widzialem chlopca, co zlapal zajaca
I-saw boy caught hare
"I saw a/the boy that caught a/the hare" [120]

(10b) *ksiazka, co ona spadla ze stolu
book it fell from table
"book that fell from a/the table" [126]

(10c) *mezczyzna co on zostal aresztowany
man he has-been arrested
"man who has been arrested" [127]

When, however, the relativized NP is in accusative or oblique case, it is impossible to *not* spell-out a case-marked pronominal in the lower clause, as indicated in (11). Example (11b) indicates the marginal possibility of avoiding pronunciation of this pronominal form when the relativized NP is inanimate. The last example, (11d), illustrates that what I've labeled the resumptive pronoun need not be immediately adjacent to 'co'.

(11a) *ten mezczyzna, co Janek widzial wczoraj
the man John saw yesterday
"the man John saw yesterday" [123a]

(11b) ?ten samochod, co Janek widzial wczoraj
the car John saw yesterday
"the car John saw yesterday" [123b]

(11c) ten chlopiec, co go widziales wczoraj
the boy him you saw yesterday
"the boy you saw yesterday" [124]

(11d) ta kobieta, co mi ja pokazywales
the woman me her you showed
"the woman who you showed me" [125]

Tableau (5) presents the marks that, by hypothesis, the candidates incur from the relevant filters. Ignore the order for now.

Tableau (5)

	RCV	Min Trace	LEC	TEL
1*[$_{CP}$ unrecoverable-wh co pronoun		*	*	*
2*[$_{CP}$ unrecoverable-wh co Ø			*	*
3*[$_{CP}$ unrecoverable-wh Ø pronoun		*	*	
4 [$_{CP}$ unrecoverable-wh Ø Ø			*	
5 [$_{CP}$ Ø co pronoun		*		*
6*[$_{CP}$ Ø co Ø	*			*
7*[$_{CP}$ Ø Ø pronoun		*	*	
8*[$_{CP}$ Ø Ø Ø	*		*	

It is clear why a three-way tie is required if both of the grammatical alternants, 5-4 and 5-5, are to surface in the same language. The wh-phrase variant, 5-4, violates only LEC. The 'co' plus resumptive variant, 5-5, violates both Minimize Trace and TEL. Allowing any of these filters to escape the tie block will be fatal because the escapee will, necessarily, eliminate one of the candidates. While it is necessary to attempt such a tie to get the desired alternation, tying constraints is far from a risk-free and guaranteed method for getting an analysis to pick out precisely the right candidates from a candidate set. It is possible to use ties to generate a system that allows multiple outputs, all of which are individually possible under some strict ranking of the constraints, but there is always a risk of allowing additional undesired outputs. Moreover, since the effects of the constraint ranking can be viewed elsewhere in the grammar, introducing a tie in order to handle data from one candidate set can have undesirable consequences throughout the rest of the system. (The present analysis actually encounters this objection, although perhaps not fatally[26]. I will have more to say about this below.) Tying two constraints results in languages whose grammatical forms constitute a subset of the union of two "strictly ranked" languages that correspond to the two different orderings of the tied constraints. The situation for either larger n-way ties or multiple tie blocks is entirely analogous.

As I will show momentarily, the three-way tie is the correct move to make for Polish. This analysis of the Polish short resumptive, then, places it and its more familiar alternant squarely in the typology predicted by the present system. As promised above, I provide the six tableaux that compute the consequences of the proposed three-way tie. In inspecting these six tableaux, it is perhaps worth considering the *a priori* likelihood of this analysis falling as neatly as it does from this constraint system, which was developed independently for the analysis of French and English[27]:

---

26 For example, in the present case, the simplest prediction would be that in Polish, as in English which also ties LE(C) and TEL, the complementizer should be optionally present in simple declaratives. This is not the case; the complementizer is mandatory in this case. I'll attempt to provide an explanation for this below.

27 Personally, this finding was quite remarkable since at some intuitive level the notion of freely allowing ties struck me as implausible; my initial efforts with the Polish paradigm involved a series of unsuccessful Procrustean attempts to fit the data to the other pattern in the taxonomy which allowed for resumptives without ties.

Tableau (6a)

	RCV>>	LEC>>	Min Trace>>	TEL
1 *unrecoverable-wh co pronoun		*!	*	*
2 *unrecoverable-wh co Ø		*!		*
3 *unrecoverable-wh Ø pronoun		*!	*	
4 unrecoverable-wh Ø Ø		*!		
5 Ø co pronoun=>			*	*
6 *Ø co Ø	*!			*
7 *Ø Ø pronoun		*!	*	
8 *Ø Ø Ø	*!	*		

Tableau (6b)

	RCV>>	LEC>>	TEL>>	Min Trace
1 *unrecoverable-wh co pronoun		*!	*	*
2 *unrecoverable-wh co Ø		*!	*	
3 *unrecoverable-wh Ø pronoun		*!		*
4 unrecoverable-wh Ø Ø		*!		
5 Ø co pronoun=>			*	*
6 *Ø co Ø	*!		*	
7 *Ø Ø pronoun		*!		*
8 *Ø Ø Ø	*!	*		

Tableau (6c)

	RCV>>	Min Trace>>	LEC>>	TEL
1 *unrecoverable-wh co pronoun		*!	*	*
2 *unrecoverable-wh co Ø			*	*!
3 *unrecoverable-wh Ø pronoun		*!	*	
4 unrecoverable-wh Ø Ø =>			*	
5 Ø co pronoun		*!		*
6 *Ø co Ø	*!			*
7 *Ø Ø pronoun		*!	*	
8 *Ø Ø Ø	*!		*	

Tableau (6d)

	RCV>>	Min Trace>>	TEL>>	LEC
1 *unrecoverable-wh co pronoun		*!	*	*
2 *unrecoverable-wh co Ø			*!	*
3 *unrecoverable-wh Ø pronoun		*!		*
4 unrecoverable-wh Ø Ø =>				*
5 Ø co pronoun		*!	*	
6 *Ø co Ø	*!		*	
7 *Ø Ø pronoun		*!		*
8 *Ø Ø Ø	*!			*

Tableau (6e)

	RCV>>	TEL>>	LEC>>	Min Trace
1 *unrecoverable-wh co pronoun		*!	*	*
2 *unrecoverable-wh co Ø		*!	*	
3 *unrecoverable-wh Ø pronoun			*	*!
4 unrecoverable-wh Ø Ø =>			*	
5 Ø co pronoun		*!		*
6 *Ø co Ø	*!	*		
7 *Ø Ø pronoun			*	*!
8 *Ø Ø Ø	*!		*	

Tableau (6f)

	RCV>>	TEL>>	Min Trace>>	LEC
1 *unrecoverable-wh co pronoun		*!	*	*
2 *unrecoverable-wh co Ø		*!		*
3 *unrecoverable-wh Ø pronoun			*!	*
4 unrecoverable-wh Ø Ø =>				*
5 Ø co pronoun		*!	*	
6 *Ø co Ø	*!	*		
7 *Ø Ø pronoun			*!	*
8 *Ø Ø Ø	*!			*

This result strengthens the argument for the current system of constraints, which have basically been taken directly from Pesetsky, though as I'll discuss below, it's not clear that there will be a direct link to the analysis of simple declaratives as there was in French and English. Moreover, the result provides further justification for adopting the reordering notion of tie. As seen above, the reordering interpretation of ties produces the right effect. If the system is to capture the desired alternation, however, the branching constraints notion of tied constraints results in an overly large set of winners, as shown in Tableau (8). The pooled constraints notion, on the other hand, can't even produce both forms involved in the alternation. This is indicated in (9).

Tableau (7): A reminder of the constraints

	RCV	LEC	Min Trace	TEL
1 *unrecoverable-wh co pronoun		*!	*	*
2 *unrecoverable-wh co Ø		*!		*
3 *unrecoverable-wh Ø pronoun		*!	*	
4 unrecoverable-wh Ø Ø		*!		
5 Ø co pronoun			*	*
6 *Ø co Ø	*!			*
7 *Ø Ø pronoun		*!	*	
8 *Ø Ø Ø	*!	*		

Tableau (8): Branching constraints notion of tie

	RCV	LEC= Min Trace= TEL
1 *unrecoverable-wh co pronoun		*!
2 *unrecoverable-wh co Ø =>		
3 *unrecoverable-wh Ø pronoun=>		
4 unrecoverable-wh Ø Ø =>		
5 Ø co pronoun=>		
6 *Ø co Ø	*!	
7 *Ø Ø pronoun=>		
8 *Ø Ø Ø	*!	

Tableau (9): Pooled violation notion of ties

	RCV	LEC= Min Trace= TEL
1 *unrecoverable-wh co pronoun		**!*
2 *unrecoverable-wh co Ø		**!
3 *unrecoverable-wh Ø pronoun		**!
4 unrecoverable-wh Ø Ø =>		*
5 Ø co pronoun		**!
6 *Ø co Ø	*!	*
7 *Ø Ø pronoun		**!
8 *Ø Ø Ø	*!	*

The same set of constraints, with one qualification, also generates the correct alternation in cases where the relativized material can recoverably delete. Tableau (10) provides a reminder of the mark-assigning behavior of the relevant constraints. The tableaux in (11) do the work of establishing the alternation between a "'co' only" form and a form where only the wh-form surfaces. They also, however, allow in the "all null" possibility, which is not attested in Polish. I'll have more to say about this shortly.

Tableau (10): The constraints

	RCV	LEC	Min Trace	TEL
1 *recoverable-wh co pronoun		*	*	*
2 *recoverable-wh co Ø		*		*
3 *recoverable-wh Ø pronoun		*	*	
4 recoverable-wh Ø Ø		*		
5 *Ø co pronoun			*	*
6 Ø co Ø				*
7 *Ø Ø pronoun		*	*	
8 *Ø Ø Ø		*		

Tableau (11a)

	RCV	LEC	Min Trace	TEL
1 *recoverable-wh co pronoun		*!	*	*
2 *recoverable-wh co Ø		*!		*
3 *recoverable-wh Ø pronoun		*!	*	
4 recoverable-wh Ø Ø		*!		
5 *Ø co pronoun			*!	*
6 Ø co Ø =>				*
7 *Ø Ø pronoun		*!	*	
8 *Ø Ø Ø		*!		

Tableau (11b)

	RCV	LEC	TEL	Min Trace
1 *recoverable-wh co pronoun		*!	*	*
2 *recoverable-wh co Ø		*!	*	
3 *recoverable-wh Ø pronoun		*!		*
4 recoverable-wh Ø Ø		*!		
5 *Ø co pronoun			*	*!
6 Ø co Ø =>			*	
7 *Ø Ø pronoun		*!		*
8 *Ø Ø Ø		*!		

Tableau (11c)

	RCV	Min Trace	LEC	TEL
1 *recoverable-wh co pronoun		*!	*	*
2 *recoverable-wh co Ø			*!	*
3 *recoverable-wh Ø pronoun		*!	*	
4 recoverable-wh Ø Ø			*!	
5 *Ø co pronoun		*!		*
6 Ø co Ø =>				*
7 *Ø Ø pronoun		*!	*	
8 *Ø Ø Ø			*!	

Tableau (11d)

	RCV	Min Trace	TEL	LEC

1 *recoverable-wh co pronoun		*!	*	*
2 *recoverable-wh co Ø			*!	*
3 *recoverable-wh Ø pronoun		*!		*
4 recoverable-wh Ø Ø =>				*
5 *Ø co pronoun		*!	*	
6 Ø co Ø			*!	
7 *Ø Ø pronoun		*!		*
8 *Ø Ø Ø =>				*

Tableau (11e)

	RCV	TEL	LEC	Min Trace
1 *recoverable-wh co pronoun		*!	*	*
2 *recoverable-wh co Ø		*!	*	
3 *recoverable-wh Ø pronoun			*	*!
4 recoverable-wh Ø Ø =>			*	
5 *Ø co pronoun		*!		*
6 Ø co Ø		*!		
7 *Ø Ø pronoun			*	*!
8 *Ø Ø Ø =>			*	

Tableau (11f)

	RCV	TEL	Min Trace	LEC
1 *recoverable-wh co pronoun		*!	*	*
2 *recoverable-wh co Ø		*!		*
3 *recoverable-wh Ø pronoun			*!	*
4 recoverable-wh Ø Ø =>				*
5 *Ø co pronoun		*!	*	
6 Ø co Ø		*!		
7 *Ø Ø pronoun			*!	*
8 *Ø Ø Ø =>				*

Before drawing broader conclusions, I'll turn briefly to Standard Russian where a similar proposal to distinguish NPs along case recoverability dimensions meets with comparable success under a different ordering of the constraints.

<u>Russian Relative Clauses</u>

Standard Russian NP wh-forms bear a striking resemblance to Polish and, therefore, it seems *prima facie* reasonable to expect to find a split between relative clause types along similar case boundaries. Data for this section comes from Comrie (1989) and Golab and Friedman (1972)

Like Polish, Standard Russian allows a class of relative clauses, shown in (12) where a case-marked wh-form is pronounced in the relative.

(12a)	devuska	kotoraja	prisla		
	girl	who-NOM	arrived [Comrie, 7.22]		
(12b)	devuska	kotoruju	ja	videl	
	girl	who-ACC	I	saw [Comrie, 7.24]	
(12c)	devuska	kotoroj	ja	dal	knigu
	girl	who-DAT	I	gave	book [Comrie, 7.26]

When either, the subject or an inanimate object NP has been relativized, an additional variant in which the wh-form is suppressed and an invariant 'cto' surfaces is also attested. Unlike Polish, Standard Russian does not allow resumptive pronouns in simple cases of relativization from non-islands. (Golab and Friedman, however, indicate this is the case in colloquial Russian, as in Polish.)

If the split between cases that participate in an alternation between a form with an overt wh- and a form with an overt 'cto' can also be assimilated to the recoverability distinction, then the same tie that works in English, modulo a similar problem with the general absence of the "all null" alternative, will work in Russian. Tableaux (12a) and (12b) present the analysis for relativization of NPs with recoverably deletable case morphology, respectively.

Tableau (12a)

	RCV>>	Min Trace>>	LE(cto)=	TEL
1*rec-wh cto pronoun		*!	*	*
2*rec-wh cto Ø			*	*!
3*rec-wh Ø pronoun		*!	*	
4 rec-wh Ø Ø =>			*	
5*Ø cto pronoun		*		*
6 Ø cto Ø =>				*
7*Ø Ø pronoun		*!	*	
8*Ø Ø Ø =>			*	

Tableau (12b)

	RCV>>	Min Trace>>	LE(cto)=	TEL
1*unrec-wh cto pronoun		*!	*	*
2*unrec-wh cto Ø			*	*!
3*unrec-wh Ø pronoun		*!	*	
4 unrec-wh Ø Ø =>			*	
5*Ø cto pronoun		*!		*
6*Ø cto Ø	*!			*
7*Ø Ø pronoun		*!	*	
8*Ø Ø Ø	*!		*	

From this point of view, Russian relative clauses satisfy the same constraint hierarchy as English relative clauses although the classes of recoverable and non-recoverable XPs crucially differ. Those NPs marked with an unrecoverable case in Russian, like English PPs, resist the leftward pressure of the complementizer. Those NPs marked with a recoverable case, because of the tie between LEC and TEL, have two options. They can either yield to leftward pressure from the complementizer, incurring a TEL violation but avoiding a LEC violation, or remain in place and force out the complementizer, avoiding a TEL violation but incurring a LEC violation. Polish relative clauses, on the other hand, provide a genuinely novel example of an ordering predicted by Pesetsky's system.

The addition of this analysis of Polish to Pesetsky's analyses of English and French, then, provides evidence for the third of five predicted patterns that fall out of the system under consideration. Currently, however, I am unaware of a language that obligatorily pronounces unrecoverable elements in resumptive position in relative clauses. Finding a clear example of this pattern of "mandatory resumption" would obviously strengthen the analysis considerably. If, as will be implicit below when modification of Tesar and Smolensky's algorithm are considered, the use of tied constraints is a marked option that developing grammars pursue only when input data from the target doesn't allow any strict ranking of constraints, then, all else being equal, one might expect to find that the preponderance of grammars can be analyzed without the use of ties. The French pattern and the mandatory resumption pattern can be generated without ties. The English pattern crucially requires a tie between LEC and TEL and the Polish pattern crucially requires adding Min Trace to this block of tied constraints. The fifth pattern can also be generated without ties by placing TEL >> LEC. With this constraint ordering, it becomes impossible to pronounce the complementizer in any position. Again, I'm not aware of well-documented cases that fit this description[28].

It also remains to be seen if the possibility of resumptives in simple relative clauses and the existence of "rich" nominal morphology will continue to correlate in an interesting way.

"Non-Optimal" Components of the Analysis

Hopefully at this point, the reader is convinced that the analysis of Polish relative clauses flows fairly smoothly from the constraint system under consideration. It is my intention now to remind the reader of three rough, but perhaps not impassable, points that I've pointed to but largely steered clear of.

First, the account incorrectly predicts the possibility in both Polish and Russian of relative clauses where neither the complementizer nor any link in the chain of a recoverable XP is pronounced. Given the four constraints under consideration, this spell-out is indistinguishable from the spell-out where only the recoverable XP in [SPEC, CP] is pronounced. Empirically,

---

[28] Pesetsky (1994) suggests giving an analysis of child language's noticeable impoverishment of functional categories in terms of the ascendancy of TEL, but much descriptive work remains to be done to flesh out this proposal.

however, only the Polish forms in (13a-b) and the Russian forms in (14a-b) are allowed. Forms like (13c) and (14c) are ungrammatical.

(13a) chlopca Ø-NP co Ø-trace zlapal zajaca

(13b) chlopca wh-form Ø-C Ø-trace zlapal zajaca

*(13c) chlopca Ø-NP Ø-C Ø-trace zlapal zajaca

'the boy that caught the hare' or 'the boy who caught the hare'

(14a) devuska Ø-NP cto Ø-trace prisla

(14b) devuska wh-form cto Ø-trace prisla

*(14c) devuska Ø-NP Ø-C Ø-trace prisla

'the girl who arrived' or 'the girl that arrived'

As the "gap" in my English glosses suggest, however, this problem is not entirely new to these Slavic languages. Pesetsky's analysis of English nominal relatives also initially predicts what I'll call the "all null" form. When the relativized element is in the accusative case, the predicted "all null" form is allowed in English. In Polish, since I've analyzed nominative case as the only recoverable case, I have no way of testing whether the "all null" form is otherwise available. In Russian, the "all null" form also fails to emerge in the recoverable inanimate accusative case. Pesetsky suggests an analysis that would derive a subject/object asymmetry in the acceptability of the "all null" form. If this line of analysis is successful it could be straightforwardly adopted for the cases of Polish and Russian nominative relatives. However, the case of Russian inanimate accusative will remain unaccounted for. Ruling out the "all null" form of the inanimate accusative relative while allowing the complementizer-only form requires either finding a reason outside of the OT system of constraints for disallowing the all null form of the inanimate accusative relative or adding a new constraint and appropriately deploying it to distinguish the "wh- only" and "all nulls" forms, currently identical from the point of view of the constraints. If the second approach is taken the filter must be able to apply to the inanimate accusative relative in Russian without applying to the accusative relative in English because allowing it to apply to the candidates for pronunciation of the accusative relative would block the desired optionality cross-linguistically. Possibly "richer" case of the Slavic sort is subject to its own edge condition and, therefore, only drops out when the complementizer surfaces at the edge of CP. This constraint could be vacuously active in the English constraint hierarchy. Alternatively, Pesetsky (personal communication) has proposed that C, unlike in English, might not be head-governed in this position in Polish and Russian. In English, the "all null" possibility must be allowed to surface in cases of object relativization; Pesetsky achieves this result by blocking the application of his ECP to simple relatives by proposing that CP is head-governed in this configuration in English. The "all null" problem could be eliminated, then, by proposing that head-government does not obtain in these configurations in Russian. This would, of course, require the maintenance of a division of cross-linguistic labor between the OT system and a more standard parametric system.

Second, while the constraint rankings provided for Polish and Russian work, modulo the paragraph above, for relative clauses. They don't work for simple declaratives. The ties between LEC and TEL would predict that, as in English, the overt complementizer is an optional component of the pronunciation of simple declaratives. However, in both languages, complementizers are mandatory in this context. There are other options to pursue, but one possibility is to simply solve the problem by fiat through the introduction of separate edge conditions for the complementizer that occurs in relatives and the complementizer that occurs in declaratives. The forms are clearly syntactically distinct, so there is no incoherence in this move. Indeed, Rizzi (1989) provides a summary morphological evidence for the claim that complementizer positions have distinct syntactic features and are sensitive precisely to the distinction between embedded declarative and relative clause environments and there are clear morphological distinctions between the different types of C in Polish and Russian. Mechanically, then, making the constraints operate over more fine-grained syntactic classes introduces no incoherence. It does, however, weaken the argument for the Optimality Analysis. The Polish and Russian analyses are, from this point of view, clearly dependent on the stronger argument for the system provided by English and French where the same morphological item serves double duty in declaratives and relative clauses. More interestingly, however, note that a variation on Pesetsky's proposed solution to the "all nulls" problem will work here as well. If, unlike in English, the heads of Polish and Russian CPs are generally not head-governed, the existence of ECP in the constraint system predicts that the complementizer will still be required even when TEL and LEC are tied.

Finally, the analyses of Polish and Russian depend on a particular, apparently morphological, notion of recoverability. The exact interpretation of recoverability is left partially open in Pesetsky (1993). Care needs to be taken that the notion of recoverability does not become entirely amorphous.

Part 2: Exploring the Learning Consequences with Ties

The introduction of ties throws a wrench directly into the workings of the CD algorithm because this algorithm is built around the assumption that if two candidates from a candidate set incur different sets of violations then both can not emerge under a single ranking of the constraints. If this assumption is not violated in the real world then hearing a particular candidate form pronounced in the target language automatically rules out any "filter-distinct" competitor; the algorithm directly exploits the indirect negative evidence provided by this situation. In this half of the paper, I will present Tesar and Smolensky's algorithm, demonstrate the problems that ties introduce and consider some possible solutions.

Tesar and Smolensky (1993): The Algorithm

Actually, the heading for this section is somewhat misleading because Tesar and Smolensky actually present a family of algorithms. However, the various members of this family are all built around the core notion of constraint demotion. For expositional purposes I will consider only their online CD algorithm. The other versions that they consider differ in certain aspects of how they organize and schedule their computations and use of input data but they are identical with respect to the class of languages they acquire in the limit.

Given that OT allows constraints to be reordered freely across languages, the learner should initially have no unrevisable commitments about ordering. In fact, learners who adopt the CD algorithm, begin with a completely unranked set, or "stratum", of constraints. From this initial state, the online CD algorithm proceeds as follows to establish a ranking. Upon encountering a piece of data from the target language the algorithm accesses the set of less optimal competitors. Each competitor is compared, in turn, with the datum from the target language. The target language datum is, by hypothesis, optimal, so the constraints must be adjusted to insure that it wins out over any competitor. Determining how the constraints must be adjusted is quite straightforward. First, the constraints that the optimal form and its competitor violate equally should be ignored; they clearly can't differentiate between the two forms. Label the remaining constraints violated by the optimal candidate *optimal violations*. Label the remaining constraints violated by the competing candidate *competitor violations*. All optimal violations must, in the target language, be ranked below a competitor violation; otherwise the optimal form would be incorrectly eliminated at some stage in the application of constraints. Optimal violations that are already in a lower stratum than some competitor require no further adjustment. In the initial state, all constraints are in the same stratum, so some readjustment is guaranteed. Optimal violations that are not already lower than some competitor, however, get placed in the stratum immediately below that of the currently highest ranked competitor violation. That is to say the problematic optimal violations are demoted just as far as they need to be to insure they can cause no problems. The algorithm creates new strata only if the currently highest ranked competitor violation turns out to actually be in the lowest stratum. This detail will turn out to be important.

I'll exemplify the algorithm's operation with a brief example. Consider a simple language characterized by the tableaux in (13):

Tableau (13a)

Candidate Set 1	Filter A>>	Filter B>>	Filter C>>	Filter D
*Form 1-1	*!			*
Form 1-2 =>		*		*

Tableau (13b)

Candidate Set 2	Filter A>>	Filter B>>	Filter C>>	Filter D
*Form 2-1		*!		
Form 2-2 =>			*	

Tableau (13c)

Candidate Set 3	Filter A>>	Filter B>>	Filter C>>	Filter D
*Form 3-1			*!	
Form 3-2 =>				*

In this system GEN takes one of three inputs and yields Candidate Set 1, 2 or 3 as appropriate. The system has four filters or constraints, A-D. In the language under consideration, where Forms 1-2, 2-2 and 3-2 are the possible output forms, the ranking A >> B >> C >> D is uniquely determined.

The online CD algorithm, of course, initially knows nothing about the correct ranking. It begins with all constraints sitting in a single stratum: {A, B, C, D}. Consider the effect on this initial representation of exposing the learner to Form 1-2. Both Form 1-2 and its competitor, Form 1-1, incur a single mark from Filter D. Therefore, exposure to Form 1-2 will not provide any useful information about the ranking of this filter.[29] However, the two forms differ with respect to Filter A—violated by 1-1—and Filter B—violated by 1-2. If Form 1-2 is to beat out Form 1-1, A must outrank B. The algorithm reflects this chain reasoning by creating a new stratum below the initial stratum and lowering B into it: {A, C, D} ->-> {B}. (I'm using a different notation for ordering in the learning algorithm to draw attention to the fact that the ordering established by constraint demotion can change in well-specified ways. For example, B will come to outrank C and D in short order.) As soon as the learner encounters Form 2-2, C must be placed below B to ensure that Form 2-1 is less optimal. The resulting representation of the learner's knowledge would be: {A, D} ->-> {B} ->-> {C}. Similarly, Form 3-2 will cause the learner to attain the final state: {A} ->-> {B} ->-> {C} ->-> {D}.

---

29 OT systems allow, and actually make use of, constraints which can assign multiple marks to a system. The way the algorithm compares competitors' ability to satisfy a constraint in these cases is to knock off marks for each candidates until one candidate has no more.

At this point, it should be intuitively plausible that the CD algorithm will, given an appropriate range of input, be generally able to establish the correct rankings for target languages analyzed in terms of standard OT systems[30]. Tesar and Smolensky rigorously establish this claim for the online CD algorithm and its ilk, with the important caveat that the learner must independently be able to take the crucial first step of generating a candidate set from a datum in the target language[31]. In the next section, I will show how the algorithm breaks down when confronted with data generated from tied constraint systems.

Tesar and Smolensky (1993): The Problem with Ties

Ties introduce an obvious mechanical difficulty. Given the method for demoting constraints, the system can never settle into a stable equilibrium.

---

30 In cases where multiple rankings would work, all strata will not be split fully. Presumably, the learner must be prepared to do something about this. Either the learner must have a way of deciding that learning is over and splitting the unresolved strata or else the learner must be able to interpret the unresolved strata as reflecting an arbitrariness of ordering that he or she is able to freely resolve. The choice that gets made here might be used to make empirical predictions about the course of acquisition.

31 The construction of this candidate-generating mechanism, however, is far from trivial, particularly in the domain of phonology. In many cases, the same surface form can be generated from different underlying forms and, therefore on the optimality view, different candidate sets. It is also far from trivial in the domain of syntax, but analyses of syntax acquisition in a number of grammatical frameworks have converged on the view that the learner is capable of inferring aspects of the meaning of a sentence without full knowledge of its syntactic structure (see for example, (Wexler & Culicover, 1980)). This, and perhaps, some partial structural analysis may be enough to cue the correct candidate set. Much more needs to be said about this issue, but I will not be saying it here.

Consider the effect of tying Filters B and C from our example above. The tableaux in (13) reflect this new grammar.

Tableau (13a)

Candidate Set 1	Filter A>>	Filter B =	Filter C>>	Filter D
*Form 1-1	*!			*
Form 1-2 =>		*		*

Tableau (13b)

Candidate Set 2	Filter A>>	Filter B =	Filter C>>	Filter D
Form 2-1 =>		*		
Form 2-2 =>			*	

Tableau (13c)

Candidate Set 3	Filter A>>	Filter B =	Filter C>>	Filter D
*Form 3-1			*!	
Form 3-2 =>				*

As a result of tying Filters B and C, two candidates that are distinct from the point of the view of the filters emerge from Candidate Set 2. Because the results of this tie are fairly easy to visualize, I haven't produced multiple tableaux. Consider how a learner equipped with the online CD algorithm would cope with an example sequence of input from this language. As before, hearing Form 1-2 would result in a transition from the innate: {A, B, C, D} to {A, C, D} ->-> {B}. From this state, hearing Form 3-2 would result in the demotion of C below D: {A, B} ->-> {C} ->-> {D}. Note, now, what happens when the learner encounters Form 2-2, one of the optimal candidates from Candidate Set 2. This leads to demotion of B below C: {A} ->-> {C} ->-> {D, B}. Eventually, of course, the learner can be expected to encounter the other alternant, Form 2-1. This results in demotion of C below B and leads to: {A} ->-> { } ->-> {D, B} ->-> {C}. Hearing 2-1 again will result in another demotion of C below B. Hearing 2-2 again, in turn, will result in demotion of B below C. It's easy to see that B and C will be continually demoted below one another. Moreover, since Form 3-2 leads directly to the ranking of D below C, it insures that filter D will also be continually entangled with filters B and C; anytime Form 3-2 is encountered D will be placed below C. More generally, any filter which ranks below a tied constraint in the target language will get caught up in a never-ending sequence of demotions.

I'm currently describing this unstoppable oscillation of constraints in fairly mechanical terms. Of course, ultimately, it is a consequence of the fact that ties violate the central logic of indirect negative evidence built into the CD algorithm. This will not prevent me from attempting to solve the problems raised by English and Polish ties by simply isolating and removing the problem of oscillating constraints. If the system is to have any chance of working, it clearly needs some additional mechanism that is able to break this oscillating pattern and introduce a tie. The hope is that, beyond this additional necessary mechanism, little else will be required. Once the ties have been identified and the appropriate tied filters have been constructed, the problem of learning the system reduces to one that the Tesar and Smolensky algorithm can solve directly. Since tied constraints can be viewed as black boxes with their own mark assigning behavior, the CD algorithm can apply directly once all the appropriate filters are in place, that is to say once the correct set of filters has been constructed.

I'll make two suggestions about how the constituent filters involved in ties might be appropriately identified and grouped. The first proposal I will make involves tweaking the system so that the tied constraints in some sense identify themselves. A slight mechanical alteration in the demotion will, for a range of languages, effectively isolate the oscillation of the tied constraints from the rest of the system. In the example above, no stable information about order could be built up for either the tied constraints or any constraints ranked below them in the target language. As stated above, the instability of the tied constraints' ordering can't be resolved without some additional mechanism. However, it may be possible, in at least some cases, for the ordering of other more highly and lowly ranked constraints to proceed uninterrupted. The second proposal that I will make attacks the problem of detecting tied constraints more directly, rather than relying passively on the ability of the system to sort itself out.

The rest of this paper, then, will be concerned with exploring modifications to the CD. This whole line of discussion, however, should be prefaced with the observation that there is one straightforward modification that can *almost* trivially be demonstrated to resolve the problem of ties.[32] Consider a version of the CD algorithm which has been augmented with a simple memory store that it uses to hold on to examples of various structures, labeled with the candidate sets that they are members of. Now, allow the algorithm to notice when it has accumulated two example structures which both come from the same candidate set. (Here the notion of "sameness" would have to be slightly abstract, referring to the syntactic categories of the constitutents. The intention is that the learner, for example, would recognize the need for a tie between LEC and TEL after hearing, for example, "the dog who I saw" and "the book that I read" because two different relative candidates from the "same" set have been revealed as possibilities.) If the two structures violate different sets of constraints, then no grammar which orders its constraints in a strict dominance hierarchy will be able to make sense of the situation. If a strict dominance hierarchy orders the constraints, only one of the structures could emerge as optimal. In a strictly ranked system, optionality is possible only along dimensions that the filters

---

32 The non-trivial part comes perhaps in determining what level of abstraction to store inputs as.

ignore. Whenever a new tie of the simple filters is detected, compose the appropriate tied filters and reinitiate the CD algorithm. As noted above, once the appropriate ties have been constructed the problem has been reduced to one that the CD algorithm can solve.

This proposal for learning ties will work, but it has the effect of conceptually undermining much of the motivation for applying OT analyses. If the learner can simply tabulate which members of a candidate set are possible productions of the target language, it is somewhat unclear what the constraint set is explaining. If the learner acquires the system by memorizing which candidates are acceptable in the target, why would it be impossible to learn that any arbitrary set of candidates is "optimal" or acceptable in a particular language? Moreover, why should the winner in a given candidate set depend on what wins in other candidate sets in the language? Given a set of constraints and a set of candidate sets, the current proposal, with no further stipulations, potentially endows the learner with the power to learn a much larger class of languages than those which can be generated by an intended OT analysis. The ability to simply record which types forms can be produced, in effect, allows the learner to simply ignore whether the resulting set of output forms is coherent from the point of view of the constraints[33].

An interesting solution to the learnability problem introduced by the possibility of tied constraints will be one which allows for the detection of ties and the composition of tied filters without requiring enough mnemonic machinery to make the OT analysis an afterthought.

---

33 The system could be made restrictive in some sense if there were a small and privileged set of candidate sets which were monitored during learning. Ties, then, could only be established on the basis of evidence from these sets.

Solution 1: Self-sorting Constraints

In some cases, a simple modification in the demotion routine in Tesar and Smolensky's algorithm will allow the algorithm to sort out a set of constraints with a single n-way tie into three correctly ordered groups of constraints: (1) a stable and correctly ordered block of constraints that dominate the constraints involved in the n-way tie, (2) an unstably ordered block of constraints consisting of the constraints involved in the n-way tie and (3) a stable and correctly ordered block of constraints that the constraints involved in the n-way tie dominate. The modification that needs to be made is a minimal one. Unmodified, the algorithm only creates new strata for the demoted constraints when the evidence requires placing a constraint below the lowest existing stratum. The new version of the algorithm will always create a new stratum when demoting constraints.

As an example, consider the problem of learning the ordering for a fuller system of constraints than I've dealt with so far: RCV, LEC, TEL and ECP from the initial Pesetsky framework[34]. As shown above, it will be impossible for the constraint demotion component of the algorithm to stably order LEC and TEL. Hearing a simple embedded declarative *with* an overt complementizer will result in TEL's demotion to a position below LEC. Similarly, hearing a simple embedded declarative *without* an overt complementizer will result in LEC's demotion below TEL. This oscillation will eventually have to be halted, but, for now, consider the fate of the other two constraints RCV and ECP[35]. Since the English system requires RCV to

---

[34] I will only consider the input forms that Pesetsky explicitly discusses. These will be sufficient in this case. In Polish, we might wonder about whether further analysis would uncover input forms that eased learnability.
[35] One could imagine a situation in which this oscillation was never resolved. A learner's grammar would flip-flop back and forth interminably

dominate all other constraints, there can be no evidence that results in RCV's demotion. For this to happen, there would have to be a datum in the target language that violated RCV and that also had a competitor that did not. Since RCV is the highest ranked constraint, this is impossible. On the other hand, there is a body of evidence in the input data that will result in the demotion of LEC, TEL and ECP to a stratum below that of RCV. Moreover, there is a datum, Form 2 in Tableau 15 below, that will result in ECP's demotion below TEL. With the modified algorithm, if TEL is below LEC when ECP gets placed below TEL, then ECP will have effectively "jumped clear" of the tie block. ECP will no longer be caught up in the ongoing oscillation between LEC and TEL. In this case, ECP is guaranteed to "jump free" in the limit.

Figure 1 provides a step-by-step trace of the algorithm's output as it operates over one possible input sequence of English data. In this example the algorithm traces out one of many possible learning paths that leads to the correct ordering of the constraints: ECP, LEC, RCV and TEL[36]. To allow the reader to follow the argument, I include tableaux taken from Pesetsky that indicate the violations incurred by embedded declarative and stacked relative candidates, but which makes no indication of a tie between LEC and TEL. Remember that the uppermost CP in a stacked relative is, by hypothesis,

---

between LE(C) >> TEL and TEL >> LE(C). When the learner became a speaker, he or she would, in turn, induce this oscillating effect in the grammars of other members of the community. This, of course, is not the situation in the real world. A speaker of English has *simultaneous* knowledge of the acceptability of simple embedded declaratives with and without the complementizer.

36 Pesetsky notes that RCV may universally be at the top of the constraint hierarchy.

head-governed and, therefore, not problematic from the point of view of the ECP constraint.

Figure (1): Example Learning Path for English CP system

Partial Ordering	Input
(1) {ECP, LEC, RCV, TEL}	embedded declarative with COMP "Paul claims that OT is connectionist"
=> (2) {ECP, LEC, RCV} >> {TEL}	embedded declarative without COMP "Alan claims it isn't"
=> (3) {ECP, RCV} >> {TEL} >> {LEC}	embedded declarative with COMP "Paul claims that it is"
=> (4) {ECP, RCV} >> { } >> {LEC} >> {TEL}	stacked relative without COMP "This is an ongoing debate between two authors that I cite who we all know"
=> (5) {RCV} >> { } >> {LEC} >> {TEL} >> {ECP}	embedded declarative with COMP "Paul claims that it is"
=> (6) {RCV} >> { } >> { } >> {TEL} >> {LEC} >> {ECP}	

Tableau (14):Embedded Declaratives

	RCV >>	LEC>>	TEL
1 Paul claims that it is =>			*
2 Paul claims Ø it is		*!	

Tableau (15): Stacked Relatives

	RCV	LEC	TEL	ECP
1 *two authors that I cite who that we all know		*	*	
2 two authors that I cite who 0 we all know		*		C*
3 two authors that I cite 0 that we all know			*	SPEC*
4 two authors that I cite 0 0 we all know		*		C*, SPEC*

A brief discussion of this example will serve to highlight some important features of both the learning problem introduced by ties and the modified Tesar and Smolensky algorithm. As soon, as the second datum is heard, it is clear from the analyst's point of view that the CD algorithm is in trouble. Two filter-distinct members of a candidate set have been heard at this point. From the more restricted point of view of the CD algorithm, however, the first sign of trouble comes when the third datum is heard and the second stratum is completely emptied. The logic of constraint demotion is such that an emptied stratum directly indicates a contradiction, if it is assumed that there are no ties.

In this example then, the constraints sort themselves out appropriately in three tiers. As noted several times above, some additional mechanism is still required to introduce a tie among the unstably, shifting constraints. If the sorting and reordering of untied constraints proceeds successfully, though, then after some amount of time the only constraints changing their order in the constraint hierarchy will be the ones that need to be tied. This would make the problem of identifying the constraints that need to be tied quite straightforward. Constraints that continue to change position after some criterial time need to be tied; the constraints that they should be tied with are the constraints that they alternate positions with. A lower bound on the amount of time the system is allowed to run before ties are resolved could be computed given some empirical estimate of the probability distribution of different sentence types in the input. In the example I've considered, it is not difficult to see that the system could manage to sort itself out quite quickly.

Several questions arise immediately: (1) Will this same type of solution work for the case of the 3-way tie that was motivated by the Polish cases discussed above? (2) Will this solution handle n-way ties generally[37]?

The answer to the first question is a very qualified yes. In fact, in the system, as I've developed so far this is almost trivially true. In the tableaux used to develop the analysis for Polish, only four constraints are considered: LEC, MIN, TEL and RCV. Provided there is evidence that will result in the demotion one of the constraints involved in the tie block to a position below RCV, then all of the constraints involved in the tie block will eventually be demoted below RCV where they will be unable to achieve a stable ordering. Very tenuously, then the solution works for this four-constraint set. What, however, would happen if the range of constraints was larger and there were more constraints below the tie block? Two problems could arise. First, the first pass sorting into above-, within- and below-tie-block sets could fail. Second, it could prove impossible to reorder the below-tie-block.

Unfortunately, the modification will not succeed in general, because there is no guarantee that constraints demoted below the tie block can be correctly reordered. Consider the language depicted in Tableaux (16a-d).

Tableau (16a)

Candidate Set 1	Filter A>>	Filter B =	Filter C>>	Filter D
Form 1-1	*			
Form 1-2 =>		*		

Tableau (16b)

Candidate Set 2	Filter A>>	Filter B =	Filter C>>	Filter D
Form 2-1	*			
Form 2-2 =>			*	

Tableau (16c)

Candidate Set 3	Filter A>>	Filter B =	Filter C>>	Filter D
Form 3-1	*			
Form 3-2 =>				*

37 One might also ask about the two-way tie introduced for Russian.

Tableau (16d)

Candidate Set 4	Filter A>>	Filter B =	Filter C>>	Filter D
Form 4-1			*	
Form 4-2 =>				*

Tableau (16e)

Candidate Set 5	Filter A>>	Filter B =	Filter C>>	Filter D
Form 5-1		*		*
Form 5-2 =>			*	

The language demonstrably requires the ranking A >> B >> C >> D. Evidence from Candidate Set 1 directly establishes A >> B. Similarly, Candidate Sets 2, 3 and 4 directly establish A >> C , A >> D and C >> D, respectively. It remains to fix B's position with respect to C and D. The outcome of the competition from Candidate Set 5 does not directly and conclusively argue for any particular ranking of filters. It does, however, provide constraints that contribute to an argument for ranking B >> C and B >> D. From Candidate Set 5, it is clear that either B >> C or D >> C. Candidate Set 4, however, has already indicated that D does not dominate C, rather C >> D. Therefore, B >> C and by transitivity, B >> D.

While the language requires A >> B >> C >> D, it is possible that a learner employing the modified CD algorithm, which always creates a new stratum to place demoted constraints in, will not acquire the correct ranking in the limit. Consider the unfortunate fate of a learner whose first exposure to the target language is the sequence in (15)

(15) Form 1-2, Form 2-2, Form 3-2.

The corresponding sequence of learning representations this individual entertained would be:

(16) initial state: {A, B, C, D}

{A, C, D} ->-> {B}

{A, D} ->-> {C} ->-> {B}

{A} ->-> {D} ->-> {C} ->-> {B}

The system has correctly inferred that A >> B; A >> C and A >> D and nothing will undo these properties of the learning representation. Data from Candidate Sets 1 through 3 is irrelevant at this point in the course of acquisition. Now, however, the learner will be unable to acquire the target. Only Form 4-2 and Form 5-2 are potentially informative at this point and, in fact, only Form 4-2 can cause a change at the outset. Initially, D is, incorrectly, in a stratum higher than C. This means that Form 5-2 can be explained without further adjustment of the constraints. Hearing Form 4-2, on the other hand, results in D's demotion below C and causes a transition to:

(17) {A} ->-> {Ø} ->-> {C} ->-> {D}->-> {B}.

Hearing Form 4-2 again, of course, would not cause further adjustments. Hearing Form 5-2 now, though, forces a change in the representation to:

(18) {A} ->-> {Ø} ->-> {Ø} ->-> {D}->-> {C} ->-> {B}

Given the modified algorithm, Form 5-2 only tells the learner to place C below D *or* B.With the exception of the two null strata this is what we began with at the end of (16). Since all possible orders of presentation of Forms 4-2 and 5-2 have been considered, it is clear that the learner can not achieve the target ranking. The proposal has created an oscillation of constraints when there is no tie.

Given Tesar and Smolensky's result, it is clear that the unmodified CD algorithm would correctly establish the ranking A>>B>>C>>D given any ordering of the stimuli from the language. This is problematic since the original CD algorithm would unproblematically handle this strictly ranked language.

The current proposal will be able to acquire English and Polish, given the analyses developed so far, but will demonstrably not handle certain possible strictly ranked grammars and may or may not break down as the analyses for English and Polish expand to include more low ranked constraints. Pinning down this portion of the argument, of course, requires a precise specification of the class of grammars in which there is potential for failure. (I use the term "potential for failure" since the failure in the acquisition of the artificial language represented in Tableaux (16a-e) depended on a particular ordering of inputs.) The reason that learning failed in the case considered above is that the relative ranking of the constraints B and D depended on exposure to several different inputs; no single datum directly indicates a need for B >> D or B >> C. Rather, Form 5-2 indicates a need for B or D >> C and Form 4-2 indicates a need for C >> D. This leads to the following chain of inference: C dominates D, therefore, D does not dominate C. Since B dominates C or D dominates C and D does not dominate C then B dominates C. By transitivity, B also dominates D. Reliance on accumulation of evidence from several pieces of data to rank constraints is problematic for the modified CD algorithm because it is possible to make a local adjustment of the constraints in response to one datum only to undo it in response to another. This insight could use a more formal statement.

<u>Solution 2: Direct Arguments for Constraint Rankings</u>

While Solution 1 might turn out to work for English and Polish, it has the undesirable property that success is not guaranteed on the full range of strict dominance hierarchies. The strategy that I'll consider now does not have this fault, although it has the converse problem of not being able to handle some of the ties handled by Solution 1, including, potentially, the tie required in Polish. The strategy adopted in Solution 2 is a simple one. Although much more restricted, it retains the central strategy of the general solution to the tie problem discussed at the outset of the second half of the paper: initiate the standard online CD algorithm, identify constraints involved in tie blocks, create a new set of filters by composing meta-filters from simple filters and then reinitiate the standard CD algorithm as necessary.

The key insight behind this maneuver is the following. A language only requires a tie if there are paradigms where filter-distinct candidates emerge as winners. These filter-distinct candidates must not differ on any constraint that is not included in the tie block. (Unless, of course, there is more than one tie in the system. This could cause problems for this proposal). If they did, then these other constraints would force selection of non-filter-distinct candidates and void the need for the tie. Consider, then what happens in a system that requires a two-way tie between two filters, A and B. There must be paradigms with at least two winners who differ only in that one violates A and the other violates B. For a learner who obeys the CD algorithm, exposure to the "A violator" directly forces the demotion of A below B, regardless of the state of the system. Similarly, exposure to the "B violator" directly forces the demotion of B below A. In a sense, then, if there were no ties hearing the "A violator" or the "B violator" would directly argue for ranking A>>B or B>>A respectively; hearing both forms would be impossible. If the learner keeps track of the constraint rankings produced by these direct arguments it will be possible to detect whether the assumption of no ties is violated. Of course, this was already possible even with the online CD algorithm; the complete emptying of a stratum directly pointed to an "inconsistency" in the input data. Detecting the need for a tie somewhere in the system, however, is not sufficient to solve the problem at hand. By keeping track of direct arguments for constraint rankings, though, it is possible to take the necessary additional step of assigning blame to the problematic constraints. Remember from our discussion of the general "solution" to the problem of ties, that identifying the tied constraints reduces the problem to one that the online CD algorithm can already solve.

Since English's tie is only two-way this logic applies directly. Given the paradigms presented so far for Polish, however, this will not help. So far, the evidence that forced the three-way Polish tie has come from a single paradigm—simple relative clauses when the relativized element bears unrecoverable case morphology. Forms where only the wh-element surfaces violate only LEC. Forms where 'co' appears with the resumptive violate both Min Trace and TEL. Hearing the 'co' plus resumptive form, then, provides direct arguments for ranking LEC above both Min Trace and TEL. Hearing the wh-form on the other hand does not provide a direct argument. Instead, it provides only a disjunctive constraint: Min Trace >> LEC or TEL >> LEC. If the three-way tie could be established pair by pair as in the language I've formulated in Tableaux (17a-c) then there would be no problem. It's not clear to me what constructions, if any, might provide the necessary evidence in Polish.

Tableau (17a)

Candidate Set 1	Filter A=	Filter B =	Filter C
Form 1-1 =>	*		
Form 1-2 =>		*	

Tableau (17b)

Candidate Set 2	Filter A=	Filter B =	Filter C
Form 2-1 =>			*
Form 2-2 =>		*	

Tableau (17c)

Candidate Set 3	Filter A=	Filter B =	Filter C
Form 3-1 =>	*		*
Form 3-2 =>		*	

Conclusion

To conclude, this paper has provided analyses of certain Polish and Russian relative clauses that place them squarely within the framework of Pesetsky's OT analysis. This analysis required an extension of the OT framework to allow for ties and several possible solutions to the learnability problems that these ties raised were considered that stopped short of a general retreat to the idea that learners accumulate a store of encountered forms. It remains to be seen if these solutions will prove to be robust.

References

Chomsky, N. (1981). Lectures on Government and Binding. Dordrecht: Foris.

Chomsky, N. (1992). A Minimalist Program for Linguistic Theory. Cambridge, MA: MIT Working Papers in Linguistics

Comrie, B. (1989). Language Universals and Linguistic Typology. Oxford: Basil Blackwell.

Fisiak, J., Lipinska-Grzegorek, M., & Zabrocki, T. (1978). An Introductory English-Polish Contrastive Grammar. Warsaw: Panstowe Widawnictwo Naukowe.

Fox, D. (1993). manuscript, MIT.

Golab, Z., & Friedman, V. (1972). "The relative clause in Slavic," in P. Peranteau, J. Levi, & G. Phares (Eds.), The Chicago Which Hunt: Papers from the Relative Clause Festival Chicago: Chicago Linguistic Society.

Goldsmith, J. (Ed.) (1993). The Last Phonological Rule: Reflections of Constraints and Derivations. Chicago: The University of Chicago Press.

Grimshaw, J. (1993). Minimal Projection, Heads and Inversion Rutgers University, New Brunswick, NJ.

Legendre, G., Raymond, W., & Smolensky, P. (1993). "Analytic typology of case marking and grammatical voice," in Proceedings of the Berkeley Linguistic Society, 19 .

Pesetsky, D. (1993). manuscript, MIT.

Prince, A., & Smolensky, P. (1993). Optimality Theory: Constraint Interaction in Generative Grammar. Rutgers University, New Brunswick; University of Colorado, Boulder.

Rizzi, L. (1990). Relativized Minimality. Cambridge, MA: MIT Press.

Tesar, B., & Smolensky, P. (1993). The Learnability of Optimality Theory: an Algorithm and Some Basic Complexity Results University of Colorado at Boulder.

Wexler, K., & Culicover, P. (1980). Formal Principles of Language Acquisition. Cambridge, MA: MIT Press.

Appendix

Resumptive in Relative Clauses Taxonomy for LEC, MIN and TEL with high RCV:

TEL>>LEC; COMPLEMENTIZER NEVER SURFACES
```
((RCV TEL LEC MIN) (REC-REL XOO OOO) (UNREC-REL XOO))
((RCV MIN TEL LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((RCV TEL MIN LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((RCV (TEL MIN) LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((RCV TEL (MIN LEC)) (REC-REL XOO OOO) (UNREC-REL XOO))
```

LEC>>TEL; MIN AT LEAST AS HIGH AS LEC; FRENCH
```
((RCV MIN LEC TEL) (REC-REL OXO) (UNREC-REL XOO))
((RCV (MIN LEC) TEL) (REC-REL OXO) (UNREC-REL XOO))
```

LEC>>TEL; LEC>>MIN
```
((RCV LEC TEL MIN) (REC-REL OXO) (UNREC-REL OXX))
((RCV LEC MIN TEL) (REC-REL OXO) (UNREC-REL OXX))
((RCV LEC (TEL MIN)) (REC-REL OXO) (UNREC-REL OXX))
```

LEC==TEL; ENGLISH (minus all-nulls subject extraction)
```
((RCV (TEL LEC) MIN) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
((RCV MIN (TEL LEC)) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
```

LEC==MIN==TEL; POLISH (minus all-nulls subject extraction)
```
((RCV (TEL MIN LEC)) (REC-REL XOO OOO OXO) (UNREC-REL XOO OXX))
```

Resumptive in Relative Clauses Taxonomy with RCV allowed to move:

TEL>>LEC; COMPLEMENTIZER NEVER SURFACES
```
((TEL MIN RCV LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((TEL LEC MIN RCV) (REC-REL XOO OOO) (UNREC-REL XOO))
((TEL LEC RCV MIN) (REC-REL XOO OOO) (UNREC-REL XOO))
((TEL RCV MIN LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((TEL RCV LEC MIN) (REC-REL XOO OOO) (UNREC-REL XOO))
((MIN TEL LEC RCV) (REC-REL XOO OOO) (UNREC-REL XOO))
((MIN TEL RCV LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((MIN RCV TEL LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((RCV TEL MIN LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((RCV TEL LEC MIN) (REC-REL XOO OOO) (UNREC-REL XOO))
(((TEL MIN) LEC RCV) (REC-REL XOO OOO) (UNREC-REL XOO))
(((TEL MIN) RCV LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
(((TEL RCV) MIN LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
(((TEL RCV) LEC MIN) (REC-REL XOO OOO) (UNREC-REL XOO))
((RCV (TEL MIN) LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((MIN (TEL RCV) LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((RCV MIN TEL LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((TEL (MIN LEC) RCV) (REC-REL XOO OOO) (UNREC-REL XOO))
((TEL RCV (MIN LEC)) (REC-REL XOO OOO) (UNREC-REL XOO))
((RCV TEL (MIN LEC)) (REC-REL XOO OOO) (UNREC-REL XOO))
(((TEL RCV) (MIN LEC)) (REC-REL XOO OOO) (UNREC-REL XOO))
((TEL (MIN RCV) LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((TEL LEC (MIN RCV)) (REC-REL XOO OOO) (UNREC-REL XOO))
(((MIN RCV) TEL LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
(((TEL MIN RCV) LEC) (REC-REL XOO OOO) (UNREC-REL XOO))
((TEL MIN (LEC RCV)) (REC-REL XOO OOO) (UNREC-REL XOO))
((TEL (LEC RCV) MIN) (REC-REL XOO OOO) (UNREC-REL XOO))
((MIN TEL (LEC RCV)) (REC-REL XOO OOO) (UNREC-REL XOO))
(((TEL MIN) (LEC RCV)) (REC-REL XOO OOO) (UNREC-REL XOO))
```

```
((TEL (MIN LEC RCV)) (REC-REL XOO OOO) (UNREC-REL XOO))
```

LEC>>TEL; MIN AT LEAST AS HIGH AS LEC; FRENCH

```
((MIN RCV LEC TEL) (REC-REL OXO) (UNREC-REL XOO))
((RCV MIN LEC TEL) (REC-REL OXO) (UNREC-REL XOO))
((RCV (MIN LEC) TEL) (REC-REL OXO) (UNREC-REL XOO))
(((MIN RCV) LEC TEL) (REC-REL OXO) (UNREC-REL XOO))
((MIN (LEC RCV) TEL) (REC-REL OXO) (UNREC-REL XOO))
(((MIN LEC RCV) TEL) (REC-REL OXO) (UNREC-REL XOO))
```

LEC==TEL; ENGLISH (minus all-nulls subject extraction)

```
(((TEL LEC) MIN RCV) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
(((TEL LEC) RCV MIN) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
((MIN (TEL LEC) RCV) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
((MIN RCV (TEL LEC)) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
((RCV (TEL LEC) MIN) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
((RCV MIN (TEL LEC)) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
(((TEL MIN LEC) RCV) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
(((TEL LEC RCV) MIN) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
(((TEL LEC) (MIN RCV)) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
(((MIN RCV) (TEL LEC)) (REC-REL XOO OOO OXO) (UNREC-REL XOO))
```

**LEC>>TEL; LEC>>RCV; MIN>>RCV; INVARIANT COMPLEMENTIZERS,
UNRECOVERABLE INFO LOST**

```
((MIN LEC TEL RCV) (REC-REL OXO) (UNREC-REL OXO))
((MIN LEC RCV TEL) (REC-REL OXO) (UNREC-REL OXO))
((LEC TEL MIN RCV) (REC-REL OXO) (UNREC-REL OXO))
((LEC MIN TEL RCV) (REC-REL OXO) (UNREC-REL OXO))
((LEC MIN RCV TEL) (REC-REL OXO) (UNREC-REL OXO))
((LEC (TEL MIN) RCV) (REC-REL OXO) (UNREC-REL OXO))
((MIN LEC (TEL RCV)) (REC-REL OXO) (UNREC-REL OXO))
((LEC MIN (TEL RCV)) (REC-REL OXO) (UNREC-REL OXO))
(((MIN LEC) TEL RCV) (REC-REL OXO) (UNREC-REL OXO))
(((MIN LEC) RCV TEL) (REC-REL OXO) (UNREC-REL OXO))
(((MIN LEC) (TEL RCV)) (REC-REL OXO) (UNREC-REL OXO))
```

**LEC>>TEL; LEC>>MIN; RCV>>MIN; COMPLEMENTIZER ALWAYS SURFACES;
RESUMPTIVE WHEN FAVORED BY RCV**

```
((LEC TEL RCV MIN) (REC-REL OXO) (UNREC-REL OXX))
((LEC RCV TEL MIN) (REC-REL OXO) (UNREC-REL OXX))
((LEC RCV MIN TEL) (REC-REL OXO) (UNREC-REL OXX))
((RCV LEC TEL MIN) (REC-REL OXO) (UNREC-REL OXX))
((RCV LEC MIN TEL) (REC-REL OXO) (UNREC-REL OXX))
((LEC RCV (TEL MIN)) (REC-REL OXO) (UNREC-REL OXX))
((RCV LEC (TEL MIN)) (REC-REL OXO) (UNREC-REL OXX))
((LEC (TEL RCV) MIN) (REC-REL OXO) (UNREC-REL OXX))
(((LEC RCV) TEL MIN) (REC-REL OXO) (UNREC-REL OXX))
(((LEC RCV) MIN TEL) (REC-REL OXO) (UNREC-REL OXX))
(((LEC RCV) (TEL MIN)) (REC-REL OXO) (UNREC-REL OXX))
```

**LEC>>TEL; LEC>>(MIN==RCV); COMPLEMENTIZER ALWAYS SURFACES;
RESUMPTIVE OPTIONALLY FOR RCV**

```
((LEC TEL (MIN RCV)) (REC-REL OXO) (UNREC-REL OXO OXX))
((LEC (MIN RCV) TEL) (REC-REL OXO) (UNREC-REL OXO OXX))
((LEC (TEL MIN RCV)) (REC-REL OXO) (UNREC-REL OXO OXX))
```

ANY REASONABLE ALTERNATIVE THAT DOESN'T PRONOUNCE THE TRACE
((MIN (TEL LEC RCV)) (REC-REL XOO OOO OXO) (UNREC-REL OXO XOO))

ANY REASONABLE ALTERNATIVE THAT SATISFIES RECOVERABILITY
((RCV (TEL MIN LEC)) (REC-REL XOO OOO OXO) (UNREC-REL XOO OXX))

4-WAY TIE
(((TEL MIN LEC RCV)) (REC-REL XOO OOO OXO) (UNREC-REL OXO XOO OXX))