

# Handling equivalence classes of Optimality-Theoretic comparative tableaux

*Igor Yanovich*  
Universität Tübingen

## ABSTRACT

Many Optimality-Theoretic tableaux contain exactly the same information, and equivalence-preserving operations on them have been an object of study for some two decades. This paper shows that several of the operations proposed in the earlier literature together are actually enough to express *any* possible equivalence-preserving transformation. Moreover, every equivalence class of comparative tableaux (equivalently, of sets of Elementary Ranking Conditions, or ERC sets) has a unique and computable normal form that can be derived using those elementary operations in polynomial time. Any equivalence-preserving operation on comparative tableaux (ERC sets) is thus computable, and normal form tableaux may therefore represent their equivalence classes without loss of generality.

*Keywords:*  
*Optimality Theory, ERC set, tableau equivalence, normal form, equivalence-preserving transformations*

Optimality Theory (OT) is a grammatical formalism based on constraint competition, formulated by Prince and Smolensky (1993) (later published as Prince and Smolensky (2004)). OT is especially popular in phonology, and is used to some extent in other branches of linguistics. In OT, a set of competing output forms  $\{Output_1, Output_2, \dots\}$  is generated by machine **Gen** for the underlying form *Input*. Each pair  $\langle Input, Output_N \rangle$  is then evaluated against a set of constraints **Con**. The grammar of a particular language is modeled as an ordering of the universal set of constraints **Con** which determines the winning input-output pair for each *Input*: an input-output pair  $\alpha = \langle Input, Output_N \rangle$  wins over another pair  $\beta = \langle Input, Output_M \rangle$  when  $\alpha$  incurs fewer vi-

relations than  $\beta$  in the most highly ranked constraint where  $\alpha$  and  $\beta$  differ. The input-output pairs that do not lose to any other pair are declared grammatical.

The OT formalism expresses two important intuitions regarding how languages might function. First, it easily captures conditions of the form “try A; if impossible, try B; if also impossible, resort to C”, which seem to frequently occur in natural language. Second, OT allows for elegant modeling of cross-linguistic variation and language change in terms of re-ranking of a universal set of constraints.

The information that a given dataset contributes constrains the possible rankings of constraints. Such information may be represented in the form of a comparative tableau (Prince 2000) or the corresponding set of Elementary Ranking Conditions, or ERC set (Prince 2002). In this paper, I present an incremental step completing the development of a full theory of equivalence classes of comparative OT tableaux, or, equivalently, ERC sets.

Earlier work, especially that of Hayes (1997), Prince (2000), Prince (2002), Brasoveanu and Prince (2011)<sup>1</sup>, and Prince (2006), has established a number of results concerning how one may transform the information in an OT tableau without loss. What has not yet been done in this line of research is to establish the limits of operations that preserve equivalence. For example, the following natural question has not been answered: given two arbitrary comparative tableaux or ERC sets, can we determine whether they contain identical information?<sup>2</sup>

The present paper fills this gap: I show that any (finite) comparative tableau may be (computably, and actually quite efficiently) transformed into a normal form, which is unique for the whole equivalence class. Moreover, this transformation is possible by applying a sequence of a set of five elementary operations and their inverses

---

<sup>1</sup> An earlier version (Brasoveanu and Prince 2005) was circulated through Rutgers Optimality Archive (ROA) <http://roa.rutgers.edu/>

<sup>2</sup> For a finite constraint set, there is only a finite number of possible rankings, so strictly speaking, brute-force testing for equivalence is possible: one may simply build every possible ranking and test whether the two tableaux/ERC sets are compatible with it. However, the number of logically possible rankings of  $n$  constraints is  $n!$ , so the complexity of brute-force testing is factorial in the number of constraints. This should be compared with the merely polynomial time complexity of our new test for equivalence through normalization given in Theorem (16).

already introduced in the literature. Only two of those are non-trivial, so a very small and simple set turns out to be sufficient to capture all the diversity of possible equivalence-preserving operations on tableaux. Normalization gives us a handle on equivalence classes of tableaux/ERC sets, as we show that each equivalence class contains exactly one normal form tableau. The normal form may therefore serve as the class's representative. A test for equivalence of arbitrary tableaux (computable for finite tableaux) involves normalizing the input tableaux and comparing the resulting normal form tableaux. The original tableaux are equivalent if and only if their normal forms are identical. Thanks to the normal form theorem proved in the present paper, the space of all possible equivalence-preserving operations may be enumerated, and the same is true of the members of which equivalence class.

1 INTRODUCTION

As a concrete example of how OT works, consider the pattern of final obstruent devoicing in Dutch.<sup>3</sup> Underlyingly, Dutch morphemes may have both voiced and voiceless obstruents: the morpheme for 'bed' is /bɛd/, surfacing faithfully in [bɛd-ən] 'beds', while the morpheme for 'dab' is /bɛt/, surfacing faithfully in [bɛt-ən] '(we) dab'. But when the final obstruent of either morpheme closes the syllable, it is realized on the surface by the same voiceless [t]: both 'bed' /bɛd/ and '(I) dab' /bɛt/ surface as [bɛt]. The following OT tableau demonstrates the violation patterns for several potential outputs corresponding to the underlying form /bɛd/:

(1)

UR: /bɛd/	*VOICED-OBS-CODA	IDENT-VOICE	*VOICED-OBS
a. [bɛd]	*		**
b. [bɛt]		*	*
c. [pɛd]	*	*	*
d. [pɛt]		**	

According to the OT conventions, solid vertical lines in the tableau indicate that the left-to-right order of the constraints corresponds to their ranking in the grammar: \*VOICED-OBS-CODA ≫ IDENT-VOICE

<sup>3</sup>My description of the Dutch pattern is based on Kager (1999).

» \*VOICED-OBS. The constraint \*VOICED-OBS penalizes any voiced obstruent. Its specialized cousin \*VOICED-OBS-CODA only penalizes voiced obstruents in the coda position of a syllable. Finally, IDENT-VOICE penalizes mismatches in voice between underlying and output consonants. The ranking in the tableau ensures that [bet] is the winning output form: [bɛd] and [pɛd] lose to [bet] in the highest constraint \*VOICED-OBS-CODA, and [pɛt] loses to it in the next constraint IDENT-VOICE. Overall, the ranking says: “avoid voiced obstruents in the coda, but preserve them elsewhere”.<sup>4</sup> We worked through this example already knowing the ranking. Normally the work of an OT analyst proceeds in the opposite direction: she would know the constraints, the violation profiles, and the designated winner, and would need to uncover the ranking that selects the winner correctly. For that procedure, it is more convenient to use a comparative OT tableau, Prince (2000). The comparative counterpart of Tableau (1) is given in Tableau (2). Each row of a comparative tableau corresponds to a pair of the winner output and one of the loser outputs of the regular OT tableau as in Tableau (1). For a specific row corresponding to a specific winner-loser pair, if the winner incurs less violations than the loser in a given constraint, the relevant cell is marked with a W; if the loser incurs less violations, the cell is marked with an L. If there is a tie, it is marked with an *e*.

(2)

UR: /bɛd/	*VOICED-OBS-CODA	IDENT-VOICE	*VOICED-OBS
[bet]~[bɛd]	W	L	W
[bet]~[pɛd]	W	<i>e</i>	<i>e</i>
[bet]~[pɛt]	<i>e</i>	W	L

It is easy to see that converting a traditional OT tableau into a comparative tableau loses information about the number of violations. But the lost information is irrelevant for recovering the ranking. Moreover, the characterization of rankings which select the correct winner becomes very simple with comparative tableaux: a ranking selects the right winner iff in every row, all L-constraints are dominated by a W-

<sup>4</sup>It is easy to check that the ranking in 1 predicts correct results for Dutch [bɛd-ən] ‘beds’, [bet-ən] ‘(we) dab’, and /bet/-[bet] ‘(I) dab’. It is also the only ranking selecting the correct winner in 1, though there exist tableaux whose winner can be correctly selected by more than one ranking.

constraint. A specific condition selecting the rankings compatible with a fixed row is called the Elementary Ranking Condition, or ERC, by Prince (2002). In Tableau (2), we can see for instance that the pair [bet]~[pet] necessitates the inclusion of a pairwise ranking IDENT-VOICE  $\gg$  \*VOICED-OBS into our grammar. On the other hand, another pair [bet]~[pɛd] does not add any useful information: without any Ls in the row, [pɛd] is going to lose to [bet] on any possible ranking of our three constraints (i.e., the ERC corresponding to this comparative row is trivial, as it is compatible with any ranking.) In what follows, I will be largely talking in terms of comparative rows and tableaux, but it is easy to translate this into talk about ERCs and ERC sets.

Turning to definitions, a **comparative tableau** is a possibly empty 2-dimensional matrix with labelled columns where each cell contains a W, an L or an *e*. The column labels of a given tableau form the **constraint set**. A **comparative row** is a comparative tableau with one row. The tableau with zero rows is special: it is compatible with any ranking whatsoever; we refer to it as  $T_{\perp}$ . A **(total) ranking** is a total order of a constraint set. In what follows, we always assume that tableaux and rankings use the same fixed constraint set.

The following terminology, mostly borrowed from Prince (2002), will also be useful. A ranking  $M$ <sup>5</sup> is **(OT-)compatible** with a comparative tableau  $T$  iff for every row, every L-constraint is dominated by some W-constraint. We say that ranking  $M$  **covers** an L in constraint  $C$  in row  $r$  when  $M$  orders one of the W-constraints of  $r$  higher than the L-constraint  $C$ . We also say that a W in any constraint  $C'$  that dominates  $C$  under ranking  $M$  **covers** the L in  $C$ . If every ranking compatible with tableau  $T$  is also compatible with tableau  $U$ , we say that  $T$  **entails**  $U$ . When  $T$  and  $U$  are compatible with exactly the same rankings, they are called **OT-equivalent**. It is trivial to extend the notions to ERC sets.

Once a comparative tableau is computed, the actual input-output pairs are no longer needed for the task of determining the correct ranking. Thus we may freely combine several tableaux stemming from different input forms into a single bigger tableau: the input informa-

---

<sup>5</sup>Prince (2002) introduces the logical perspective on OT compatibility wherein rows/ERCs are formulas, and rankings are essentially models. Hence  $M, N$  as designations for rankings.

tion in it may be viewed as being about the grammar of the language rather than about particular linguistic forms. In this paper, we will be working exclusively with comparative tableaux.

Tableaux directly computed from particular linguistic forms are often suboptimal in how they represent information. For example, the second row of 2 may be omitted without any loss of information; similarly, the W in \*VOICED-OBS in the first row is “false”, because replacing it with an *e* will not change which ranking selects the right winner. It thus becomes important to study equivalence relations between comparative tableaux/ERC sets. To name just a few examples, Hayes (1997) (cf. also a follow-up in Prince (2006)) seeks to find transformations for tableaux allowing for better information extraction; Prince (2000) introduces the notion of entailment between rows and tableaux; Brasoveanu and Prince (2011) define an algorithm transforming an arbitrary tableau into a small-size “basis” conveniently representing the same information.

The current paper continues that line of investigation. Namely, I prove that the equivalence-preserving operations introduced in the earlier literature are already enough to handle equivalence classes of comparative tableaux/ERC sets, once we add the necessary proofs. By definition, any (comparative) tableau *T* belongs to an equivalence class  $\mathcal{C}$  such that any tableau in  $\mathcal{C}$  is compatible with exactly the same rankings. Whenever there are such non-trivial equivalence classes, there is a problem of handling them: in geometry, there are congruence classes of geometrical figures; in proof theory, there often exist many proofs of the same statement; in lambda-calculus, there are plenty of equivalent lambda-terms. In all those cases we want to be able to obtain results common for the equivalence class. Our strategy for getting a handle on equivalent classes of OT tableaux will be fairly standard: we will find a special representative which exists in every equivalence class, and is unique in it — in other words, a normal form that can represent the class.

The plan is as follows. In Section 2, I review several elementary equivalence-preserving transformations of tableaux from the earlier literature, adding their inverses where needed. Later it will be shown that the introduced set of operations is functionally complete (that is, any equivalence-preserving transformation can be decomposed into a sequence of elementary transformations from the set). In Section 3 I

define a normal form for OT tableaux, and prove the central result of the paper: a normal form is unique in its equivalence class. This means that the normal form may be used as the representative of a class, or its *name*. Finally, in Section 4 I provide several easy corollaries following from the normal form theorems. For example, we obtain a test of equivalence for OT tableaux, and a proof that bases of Brasoveanu and Prince (2011) are unique in their equivalence classes and thus can serve as class representatives (just as normal-form tableaux can).

2

## FIVE ELEMENTARY EQUIVALENCE-PRESERVING TRANSFORMATIONS

In this section, we provide the definitions for five operations with inverses that will be shown in the next section to form a functionally complete set. The operations are either trivial (Operations (3) and (4)) or have been described and proven correct before (Operations (5) and (6) are either explicitly discussed by, or immediately follow from Prince (2002); Operation (7) is studied in Prince (2006)). The proofs of equivalence-preservation are provided here mainly for completeness' sake, so the readers familiar with the operations may wish to skip them. The novelty of the present paper is not in the operations themselves, but in the fact that together they form a functionally complete set that is enough to represent any possible equivalence-preserving operation whatsoever.

The order of columns in example tableaux below does *not* correspond to any ranking, unlike in the previous section.<sup>6</sup> Constraint names are chosen to be  $C_1, C_2, \dots$ , rather than the usual meaningful names, to underscore the fact that the transformations are completely blind to actual linguistic content, and only concern the formal information encoded in a tableau.

We use variables  $M, N, \dots$  for OT rankings; variables  $T, U, \dots$  for comparative OT tableaux; and  $r$  and  $q$  for comparative OT rows.  $W(r)$ , for row  $r$ , denotes the set of constraints that have a  $W$  in  $r$ . Similarly for  $L(r)$ . This short notation allows us to define new rows compactly: e.g.,

---

<sup>6</sup>Sometimes the absence of order is marked by using dashed vertical lines. We refrain from this practice at the request of a reviewer.

if we say that  $W(r) = \{C3\}$  and  $L(r) = \{C1\}$ , and CON is the 5-constraint set  $\{C1, C2, C3, C4, C5\}$ , then row  $r$  is the row  $(L, e, W, e, e)$ .

The first two operations we will consider are trivial. First, row swaps defined in Operation (3) never affect OT-equivalence, as the order of the rows is not significant for determining whether a ranking  $M$  is compatible with the tableau. (If we think in terms of corresponding ERC sets, the very concept of row order becomes irrelevant.) Row swap is its all inverse. Second, if a tableau is not compatible with any ranking whatsoever (that is, if it puts contradictory requirements on the ranking of constraints), there is no useful information in it anyway, so as long as the tableau remains contradictory, any changes to it do not offend equivalence (Operation (4)).

- (3) **Row swaps:** swapping any two rows preserves OT equivalence.

C1	C2	C3	C4
e	W	L	L
W	e	L	e

↔

C1	C2	C3	C4
W	e	L	e
e	W	L	L

*Proof:* trivial.

- (4) **Contradictory jumps:** for a contradictory tableau (that is, a tableau not compatible with any ranking), any row can be added, or, inversely, subtracted as long as the resulting tableau is still contradictory.

C1	C2	C3	C4
W	L	e	e
L	W	e	e

↔

C1	C2	C3	C4
W	L	e	e
L	W	e	e
e	e	W	L

*Proof:* trivial.

Row splitting and its inverse, row merging, are also nearly trivial. Given the ERC theory of Prince (2002), it is easy to show that a row with several Ls is equivalent to a set of single-L rows. In ERC terms, such single-L rows have been called Primitive Ranking Conditions by Prince (2006, p. 4). The correctness of row splitting and row merging shows that covering each L in a multiple-L row is independent from covering the other Ls. Working with single-L rows, or PRCs, is often



more convenient, especially when we turn all rows in a tableau into this single-L/PRC form.

- (5) **Row splittings and mergings:** a row  $r$  is equivalent to any set of rows  $r_1, \dots, r_n$  such that  $\forall r_i : W(r_i) = W(r)$ , and  $\bigcup_i L(r_i) = L(r)$ . That is,  $r, r_1, \dots, r_n$  must have exactly the same Ws, and the combined Ls of  $r_1, \dots, r_n$  must form the same set as the Ls of  $r$ .

C1	C2	C3	C4
W	W	L	L

↔

C1	C2	C3	C4
W	W	L	e
W	W	e	L

*Proof:* Suppose a ranking  $M$  puts on top of each L in  $r$  one of  $r$ 's Ws. As any  $r_i$  has the same Ws, any L in any  $r_i$  will also be covered by a W under ranking  $M$ .

Conversely, suppose a ranking  $N$  is compatible with all rows  $r_1, \dots, r_n$ . Consider some L of row  $r$ . Some  $r_i$  must have an L in the same constraint, and ranking  $N$  covers it with a W in one of  $W(r_i)$ . That W-constraint in  $r_i$  also has a W in  $r$ , by definition. Thus  $N$  covers the arbitrary L in  $r$  just as well.

Thus a ranking is compatible with  $r$  iff it is compatible with  $r_1, \dots, r_n$ . □

The remaining two pairs of operations are the non-trivial part of the set. Some OT rows may be superfluous in their tableaux: even if we delete them, the amount of information in the tableau does not change (e.g., the second row in Tableau (2) is superfluous.) By definition, subtraction or addition of such rows does not offend OT equivalence. What is non-trivial, though, is determining the exact formal conditions under which a row is superfluous. In the proof, I use the criterion by Prince (2002), featuring his operation of fusion.<sup>7</sup> One can provide an alternative characterization of superfluousness based on

---

<sup>7</sup> The operation of fusion on rows is defined by Prince (2002, page 8, Equation (12)). For tableau  $U$ , the fusion row  $fU$  has an  $e$  in the  $C_i$  cell iff all rows in  $U$  have an  $e$  in  $C_i$ ; has an L iff some row in  $U$  has an L in  $C_i$ ; and has a W otherwise, that is, when at least one row in  $U$  has a W in  $C_i$ , and all other rows have either Ws or es, but not Ls.

domination chains of constraints, but the proof based on such chains is more cumbersome.<sup>8</sup>

Using the fact that row order is not significant in a tableau, cf. Operation (3), we can safely use set notation for tableaux, understood as being parasitic on the notation for ERC sets:  $T \setminus r$  denotes tableau  $T$  with row  $r$  subtracted;  $T \cup U$  is a concatenation of tableaux  $T$  and  $U$ ; and so forth.

(6) **Inference eliminations and introductions:** a row  $r$  entailed in tableau  $T$  by the rest of the tableau (that is, by  $T \setminus r$ ) can be subtracted from  $T$ , or added back to tableau  $T \setminus r$ .

C1	C2	C3	C4
W	L	e	e
e	W	L	e
W	e	L	e

↔

C1	C2	C3	C4
W	L	e	e
e	W	L	e

*Proof:* Trivial. What is non-trivial is how to determine if  $r$  is entailed by  $T \setminus r$ . By Prop. 2.5 of (Prince 2002, p. 14),  $r$  is entailed by  $T \setminus r$  iff there exists a subtableau  $U$  of  $T \setminus r$  s.t. the fusion  $q$  (cf. Footnote (7)) of  $U$  entails  $r$ . In turn,  $q$  entails  $r$  either when  $r$  has no L-s and thus is compatible with any ranking, or when  $W(q) \subseteq W(r)$  and  $L(q) \supseteq L(r)$ . □

---

<sup>8</sup>I provide the definitions of possible and maximal domination chains in (i), and the criterion of superfluousness based on them, without proof, in (ii):

- (i) For a tableau  $T$ , a row  $r_i \in T$ , and a  $C_j \in L(r_i)$ , a **possible domination chain** is a sequence of constraints  $\langle C_{k_1}, \dots, C_{k_n} \rangle$  s.t.  $C_{k_n} = C_j$ , a single constraint never occurs twice in the chain, and for each  $C_{k_i}, C_{k_{i+1}}$  there is a row  $r_m \in T$  for which  $C_{k_i} \in W(r_m), C_{k_{i+1}} \in L(r_m)$ . A **maximal possible domination chain** is a possible domination chain for which there is no  $r_m \in T$  s.t.  $C_{k_i} \in L(r_m)$ .
- (ii) **Superfluous row theorem.** A tableau  $T = \langle r_1, \dots, r_n \rangle$  entails a row  $q$  iff for each  $C_i \in L(q)$ , there exists such a row  $r \in T$  in every maximal domination chain for  $C_i, r$ , and  $T$ , that there is a constraint  $C_{k_i}$  in it s.t.  $C_{k_i} \in W(q)$ .

Checking the criterion based on maximal chains does not require computing new rows, as the fusion criterion does. But it is easy to see from the cumbersome of the definitions that proving the criterion's correctness from first principles requires a bit of work. Therefore I simply reuse Prince's fusion-based result in the main text, referring the reader to Prince (2002) for proofs of its correctness.

To reduce the computational complexity of inference elimination, an RCD-based method is proposed by (Prince 2002, Sec. 5). Prince shows that instead of checking the fusions of all subtableaux, one may check whether  $T \setminus r$  is consistent with the negative  $\neg r$  of  $r$ , obtained by replacing all  $r$ 's Ws with Ls and vice versa. For  $m$  rows, we need  $m$  such RCD-based checks. As Magri (2009) explains, RCD requires  $m^2n$  operations for a tableau with  $m$  rows and  $n$  constraints. The complexity of RCD-based inference elimination is thus polynomial, in contrast to subtableau-fusion version which is exponential in the number of constraints  $n$ .

Finally, not all Ws in an OT tableau are necessarily equal: there may be rows with “false Ws” such that there is no ranking compatible with the tableau which puts that W on top of any Ls in the row. As shown by (Prince 2006, p. 12), such false Ws may be replaced with an  $e$  without affecting the set of rankings the tableau is compatible with. An example of such a W is the W in the first row in C3 in the left tableau in Operation (7). The third row of the tableau necessitates ordering C4 over C3 in any compatible ranking  $M$ , and because of that the L in the first row may never be covered by the W in C3 in  $M$ . Therefore replacing that W with an  $e$ , as in the right tableau, does not offend OT-equivalence. The operation for doing such changes is called **Generalized Removal of W**, or GRW. We also introduce its inverse, **Generalized Introduction of W**, or GIW.

- (7) **Generalized Removal of W (GRW) and Introduction of W (GIW)**: informally, a “false” W is a W whose replacement with an  $e$  does not change which rankings the tableau is compatible with. Thus a false W does not do any actual work. The example tableaux below may help visualize the phenomenon.

C1	C2	C3	C4
W	W	W	L
W	L	$e$	$e$
$e$	$e$	L	W

↔

C1	C2	C3	C4
W	W	$e$	L
W	L	$e$	$e$
$e$	$e$	L	W

Turning to the formal definition: for rows  $r$  and  $r'$  such that instead of  $r$ 's W in a fixed  $Ci$ , row  $r'$  has an  $e$ , consider a pair of  $T$  including  $r$  that is not entailed by the rest of  $T$ , and  $T' := (T \setminus r) \cup r'$ .

(That is,  $T'$  that is exactly like  $T$ , but with the  $W$  in  $C_i$  in row  $r$  replaced with an  $e$ .)

The claim is:  $T$  and  $T'$  thus defined are equivalent iff  $T \setminus r$  entails the row  $q$  such that  $W(q) := L(r) \cup (W(r) \setminus C_i)$ , and  $L(q) = \{C_i\}$ .

*Proof.* Just as with inference eliminations, the fact that a false  $W$  can be replaced with an  $e$  is trivial. What is non-trivial is the criterion for false  $W$ s: a  $W$  is false iff the row  $q$  as described above is entailed by  $T \setminus r$ . Prince (2006, p. 12) proves essentially that criterion in his (31) using fusion.<sup>9</sup> Yanovich (2011) provides a different proof in his (125) using partial OT rankings. The proof below is based on the idea of the proof in Yanovich (2011), but does not use either fusion or the apparatus of partial rankings.

Consider row  $r$  with a  $W$  in  $C_i$ , and row  $q$  defined as in the criterion above:  $W(q)$  contains all  $W$ - and  $L$ -constraints of  $r$  except  $C_i$ , and the only  $L$ -constraint of  $q$  is  $C_i$ . We need to show that the  $W$  in  $C_i$  in  $r$  is false in tableau  $T$  precisely when the rest of the tableau,  $T \setminus r$ , entails the row  $q$  so constructed.

Without loss of generality, assume that  $r$  has only one  $L$ , in constraint  $C_j$ . (We have the right to assume that because we proved in 5 that any multiple- $L$  row may be split into several single- $L$  rows that are together equivalent to it.)

Suppose  $q$  is entailed by  $T \setminus r$ . We will prove that  $T$  is then equivalent to  $T'$ , and thus the  $W$   $C_i$  in  $r$  is false. Assume towards a contradiction that there is a ranking  $M$  which is compatible with  $T$ , but not with  $T'$ . That ranking  $M$  must be compatible with  $r$ , but not with  $r'$  which differs from it in that it has an  $e$  in  $C_i$  instead of a  $W$ . Then  $M$  must say that  $C_i \gg C_j$  and that for every  $C_k$  from  $W(r) \setminus C_i$ ,  $C_j \gg C_k$ : otherwise it would be compatible not only with  $r$ , but also with  $r'$ . But then  $M$  is incompatible with  $q$ : the  $L$ -constraint  $C_i$  dominates  $C_j$  in  $M$ , and then by transitivity any  $W$ -constraint  $C_k$ . This is contrary to assumption, and therefore there cannot be such an  $M$ . Furthermore, any ranking compatible with  $T'$  is bound to be compatible with  $T$ ,

---

<sup>9</sup>Prince's theorem is slightly weaker compared to our formulation: Prince requires all rows in  $T$  to be not entailed by the rest of the tableau. His actual proof, though, only employs the fact that  $r$  is not entailed by  $T \setminus r$ , just as our proof does.

and thus we derive that if  $q$  is entailed by  $T \setminus r$ , then  $T$  and  $T'$  are OT-equivalent.

For the other direction, suppose  $q$  is not entailed by  $T \setminus r$ . We show that then there is a ranking compatible with  $T$ , but not  $T'$ . We need to show that there exists ranking  $M$  compatible with  $T \setminus r$  that says  $C_i \gg C_j \gg C_k$  for all  $C_k \in W(r')$ : such a ranking will be compatible with  $r$ , but not with  $r'$ . Towards a contradiction, suppose there is no such  $M$ . That is only possible if no ranking compatible with  $T \setminus r$  says  $C_i \gg C_j \gg C_k$ . For the  $C_i \gg C_j$  part,  $T \setminus r$  cannot necessitate the opposite ordering  $C_j \gg C_i$ : if it did, then it would have entailed  $q$ , contrary to assumption. For  $C_j \gg C_k$ , suppose towards a contradiction that every ranking compatible with  $T \setminus r$  says for some  $C_k \in W(r')$  or other that  $C_k \gg C_j$ . That can only be if there is a row  $s$  in  $T \setminus r$  with an L in  $C_j$ , and  $W(s) \subseteq W(r')$ . But if that is so, then the row  $s$ , and thus  $T \setminus r$  as a whole, entail  $r$ : the L is in the same place in  $s$  and  $r$ , and  $W(s) \subseteq W(r') \subset W(r)$ . That is contrary to assumption, so if  $T \setminus r$  does not entail either  $q$  or  $r$ , then there must be a ranking  $M$  compatible with  $T \setminus r$  saying  $C_i \gg C_j \gg C_k$  for all  $C_k \in W(r')$ . That  $M$  is compatible with  $T$ , but not with  $T'$ , and thus witnesses that  $T$  and  $T'$  are not OT-equivalent:  $T$  is compatible with a larger number of rankings, thanks to the non-false  $W$  for which the criterion based on a specially constructed row  $q$  fails.  $\square$

We have now defined and proved correctness of five pairs of elementary operations preserving OT-equivalence of comparative tableaux. Those operations as such have been known before. What has not been known is that those five pairs form a *functionally complete* set: any transformation preserving OT-equivalence can be performed by applying a sequence of those elementary operations, as we will show in the next section.

The following easy-to-prove fact will become useful later:

- (8) All operations in (3)–(7) have inverses: row swap is self-inverse; for the other four pairs, the two members of the pair are inverses.

What (8) means is that each sequence of applications of our elementary operations may be inverted: if we can derive from tableau  $T$  another tableau  $U$  using those operations, then we can also derive

from  $U$  the original tableau  $T$  by applying the inverted form of the same sequence.

### 3 NORMAL FORM FOR OT TABLEAUX

In this section, we present core novel results of this paper: two theorems regarding the existence and uniqueness of normal form for comparative OT tableaux. Namely, we define a specific tableau format in Definition (9), and then prove that for each equivalence class  $\mathcal{C}$ , there exists exactly one tableau in such format, and moreover, that the normal form of a (finite) tableau is computable. Normal forms thus can serve as true representatives of their equivalence classes, giving us a handle on those.

It should be stressed that there is nothing particularly special about normal forms — in fact, as we will see in the next section, other forms may be proven to be usable as normal forms just as well. The reason we define the normal form in Definition (9) the way we do is simply that it is convenient for proof purposes. Nor is the form we chose new: Prince (2006, p. 6) defines essentially the same form in terms of ERCs, called the Minimal Primitive Generator, or MPG. Thus in this section we show that (the tableau counterpart of) an MPG is a true normal form for OT equivalence classes.

#### (9) Normal form for OT tableaux:

1. The only contradictory tableau in the normal form is the one-row tableau with a single L in the first constraint. We can refer to this special tableau as  $T_{\perp}$ .
2. Each row has at most a single L.<sup>10</sup>
3. There are no rows which can be inference-eliminated (see Operation (6)).
4. In multiple-W rows, there are no false Ws (see Operation (7)).

---

<sup>10</sup>Such single-L rows correspond to Primitive Ranking Conditions of Prince (2006).

5. The rows are ordered according to some strict total order of the set of all possible rows.<sup>11</sup> (For corresponding ERC sets, the notion of row order becomes irrelevant.)

Here is an example of a normal form tableau:

(10)

C1	C2	C3	C4	C5
W	e	L	e	e
W	e	e	L	e
W	e	e	e	L
e	W	e	L	e

Just calling something a normal form does not make it one. The results in (11)–(14) establish the fact that the class of tableaux defined in Definition (9) indeed has normal form properties.

(11) **Normal Form Existence Theorem**

An arbitrary (finite) tableau  $T$  can be transformed into an equivalent normal form tableau by a (finite) sequence of equivalence-preserving transformations in Operations (3)–(7).

- (12) **Corollary to Theorem (11).** Each non-empty equivalence class of tableaux contains at least one normal form tableau.

*Proof of Theorem (11).* We give an explicit procedure for transforming an arbitrary tableau so that it satisfies the requirements in Definition (9). For contradictory tableaux, we just add the row (L, e, e, ...), and subtract all others. If the tableau is not contradictory, we apply row splittings until all rows have at most one L (and are thus PRC-rows). Assuming the tableau is finite, we can eliminate all entailed rows by testing whether the fusions of subtableaux satisfy Prince’s condition on entailment, see Operation (6). After that, we can similarly eliminate all false Ws from the resulting tableau by testing if the conditions for GRW, see Operation (7), are met (as all entailed rows were eliminated by that point, the row independence precondition of the criterion in Operation (7) is met). We finish the procedure by applying row swaps to get the ordering right. □

<sup>11</sup> The actual choice of ordering is irrelevant as long as all conceivable rows are strictly ordered. I will use the following: 1) let the first constraint where only one of  $r$  and  $q$  has a W be  $C_i$ ; then the row with the W in  $C_i$  goes first; 2) for rows which have identical W-sets, the row which has an L in the first constraint where only one of them has an L goes first.

*Proof of Corollary (12).* Trivial: if there were no normal form tableau in a non-empty equivalence class, then Theorem (11) could not have been valid.  $\square$

Note that if the tableau is finite, the normalization procedure described in our proof of Theorem (11) is computable. This is important because if the normal form were not computable, we could not use it without restrictions in place of any other tableau in its equivalence class: we would not have been able to ensure we can actually derive one from the other in a finite amount of time. In fact, complexity analysis shows that normalization is not only computable, but quite efficient:

- (13) Tableau normalization as defined in the proof of Theorem (11) runs in time polynomial in the number of rows  $m$  and the number of constraints  $n$ .

*Proof of Theorem (13).* Consider tableau  $T$  with  $m$  rows and  $n$  constraints. Consistency check may be performed through fusing all subtableaux of  $T$  and checking if any resulting fused row has only Ls — or equivalently and faster using RCD, as shown by Prince (2002, Section 4). To perform RCD, we need  $m^3n$  operations (Magri 2009, p. 371). Next, we do row splittings, which for any of the  $m$  rows cannot result in creating more than  $n$  rows of  $n$  constraints each, so this requires at most  $mn^2$  operations. The number of rows in the resulting split tableau is not greater than  $mn$ . Next, we check for entailed rows to eliminate. As we discussed above regarding Operation (6), rather than doing subtableau fusion, exponential in the number of rows  $m$ , we can do instead  $m$  RCD-based checks as described by Prince (2002, Section 5). We have  $mn$  rows, and each RCD involves  $(mn)^3n$  operations, so overall we need  $m^3n^4$  operations for this step. Finally, we need to check for false Ws. For that we check every W in  $mn$  rows, so at most this would be  $mn^2$  checks (actually, much less, as the same row cannot contain both  $n$  Ls and  $n$  Ws, but we can ignore this.) Each test involves checking whether the rest of the tableau entails a specially constructed row for each particular W. Again, the cost of an entailment check for a single row and a tableau with  $mn$  rows is  $m^3n^4$ , so overall we have at most  $m^4n^6$  operations. This will be the dominating term



in our complexity estimate. The time complexity of normalization is thus polynomial, which is very good.  $\square$

From Theorems (11) and (12), we know that each equivalence class has at least one normal form tableau. But can a class contain more than one normal form? Theorem (14) shows that it cannot, and thus a normal form tableau *defines* its class: it is its unique representative. To prove that fact, we will need to use relatively complex ranking-construction techniques.

**(14) Normal Form Uniqueness Theorem**

In each equivalence class of OT tableaux, there is at most one normal form tableau.

*Proof of Theorem (14).* We show that any two distinct normal form tableaux  $T$  and  $U$  belong to different equivalence classes.

Pick some row  $r$  from  $T$  which is not shared by  $U$  (in case  $T \subset U$ , we immediately derive the conclusion by considering a row from  $U$  that is not in  $T$ , and the fact that  $T$  cannot entail that row). Either our pick  $r$  is entailed by  $U$ , or it is not. In case  $r$  is not entailed by  $U$ , there is some ranking  $M$  compatible with  $U$ , but not with  $r$ , and thus not with  $T$ , so  $U$  and  $T$  are not OT-equivalent.

The interesting case is when  $U$  entails the row  $r$  we picked. We will show that in that case, there must be some ranking compatible with  $T$ , but not with  $U$ . We pick a minimal subtableau  $V$  of  $U$  that still entails  $r$ . As  $V$  entails  $r$ , every ranking compatible with  $V$  must also put one of  $r$ 's  $W$ s on top of  $r$ 's  $L$ . That can only be if there is a row  $q \in V$  which has an  $L$  in the same constraint where  $r$  has an  $L$ . Let's call that constraint  $C_i$ .

Suppose towards contradiction that  $T$  and  $U$  are equivalent, that is, compatible with exactly the same rankings. Consider some  $M$  compatible with  $T$ , and accounting for  $V$  "in the minimal possible manner": let  $M$  contain the domination chain  $Ck_1 \gg Ck_2 \gg \dots \gg C_i$  where each pairwise ranking  $Ck_1 \gg Ck_2, \dots, Ck_n \gg C_i$  accounts for one of the rows in  $V$ , but no other pairwise rankings accounting for any of  $V$ 's rows. As  $V$  is in normal form and all its  $W$ s are not false, it must be possible to construct such an  $M$ . As  $V$  entails  $r$ , constraint  $Ck_1$  is a  $W$ -constraint in  $r$ .

$T \setminus r$  cannot entail  $V$ : if it did, it would have entailed  $r$  by transitivity, which is contrary to the normal form assumption. Therefore it must be possible to lower one of  $Ck_i$  constraints below  $Ci$  building a ranking  $M'$  which is still compatible with  $T \setminus r$ , but not with  $V$ . As  $V$  entails  $r$  by assumption,  $M'$  must also be incompatible with  $r$ . But that can only be if the lowered constraint has to be  $Ck_1$ , a  $W$ -constraint in  $r$ , for otherwise  $M'$  would have still said  $Ck_1 \gg Ci$ .

We modify  $M'$  as follows: raise  $Ck_1$  just on top of  $Ci$ , but below  $Ck_n$ , resulting in  $M'' = \dots Ck_2 \gg Ck_3 \gg \dots \gg Ck_n \gg Ck_1 \gg Ci$ . That ranking  $M''$  is incompatible with  $V$ , because by construction there must have been a row in  $V$  for which we needed the pairwise ranking  $Ck_1 \gg Ck_2$ , and  $M''$  says  $Ck_2 \gg Ck_1$ . But at the same time  $M''$  is compatible with  $r$ , as it puts one of its  $W$ s on top of its  $L$ . Now compare  $M'$  and  $M''$ , and consider their compatibility with  $T \setminus r$ .  $M'$  was compatible with  $T \setminus r$ .  $M''$  differs from it in that it says  $Ck_1 \gg Ci$  instead of  $Ci \gg Ck_1$ . That change could not make  $M''$  incompatible with  $T \setminus r$ : the initial ranking  $M$  also said  $Ck_1 \gg Ci$  and was compatible with  $T \setminus r$ . Therefore we have built a ranking, namely  $M''$ , which is compatible with  $T \setminus r$  and with  $r$ , but not with  $V$ . This ranking witnesses that  $T$  and  $U$  are not equivalent.  $\square$

Theorems (11) and (14) together entail that there is exactly one normal form tableau per equivalence class. Thus a tableau as described in Definition (9) is a true normal form: a full-fledged representative, or a “name”, of its equivalence class.

In practical terms, that means that in our proofs, we can capitalize on the many nice properties of normal forms, knowing that the results will generalize to arbitrary tableaux. In the next section, we illustrate that the use of the normal form results in several simple corollaries.

#### 4 CAPITALIZING ON THE NORMAL FORM RESULTS

Theorems in (15), (16) and (18) serve two purposes. First, they have independent value, especially the proof that Brasoveanu and Prince’s SKB bases are unique in their equivalence classes. Second, the proofs of these statements illustrate how one can use the normal form results in practice to handle equivalence classes of OT tableaux.

- (15) Operations (3)–(7) form a *functionally complete set*: any tableau can be transformed into any equivalent tableau by a sequence of such operations.

*Proof of (15).* By Theorems (11) and (14), any pair of equivalent tableaux may be transformed into the same normal form tableau by a sequence of operations in Operations (3)–(7). To conclude the proof, we observe that an inverted sequence transforms the normal form back into the original tableau. By normalizing the first tableau, and then denormalizing it by applying the inverted sequence built for the second tableau, we transform the first tableau into the second.  $\square$

- (16) Equivalence of finite OT tableaux is computable in polynomial time.

*Proof of (16).* To test tableaux  $T$  and  $U$  for equivalence, it suffices to normalize both and check whether the resulting normal forms are the same. All operations are computable, for finite tableaux.

The complexity of this test is polynomial: by Theorem (13), the time complexity of normalization is polynomial in the number of rows  $m$  and the number of constraints  $n$ , and we need two such normalizations, plus a comparison of two resulting normal form tableaux which is also polynomial in  $m$  and  $n$  for the original tableaux. This fairly moderate complexity may be compared with the enormous factorial complexity of the brute-force test for equivalence that involves testing every possible ranking for compatibility with each tableau, cf. Footnote 2.  $\square$

Brasoveanu and Prince (2011) define a dense format of tableaux called the Skeletal Basis (SKB) and an algorithm turning an arbitrary tableau into an equivalent tableau in that format. An SKB of tableau  $T$  is a tableau  $T'$  such that 1) there is no OT-equivalent tableau with a smaller number of rows; and 2) no other equivalent tableau of the same cardinality has more *es*. Tableau (17) is the Skeletal Basis of the normal form tableau in Tableau (10):

(17)

C1	C2	C3	C4	C5
W	<i>e</i>	L	L	L
<i>e</i>	W	<i>e</i>	L	<i>e</i>

Brasoveanu and Prince (2011) claim to have proven, in an unpublished manuscript, the fact that *for a single tableau*, the SKB basis is unique. (Prince (2006, page 6) derives from that the result that MPGs, corresponding to our normal forms, are also unique for a single tableau.) Using our uniqueness theorem for normal forms in Theorem (14), we prove a much stronger result for SKBs in Theorem (18): each *equivalence class of tableaux* has a unique SKB.

(18) Each equivalence class of OT tableaux has exactly one tableau in the Skeletal Basis (SKB) form of Brasoveanu and Prince (2011).

*Proof of (18).* By showing that SKBs are in one-one correspondence with normal forms.

If we apply all possible row mergers to a normal form tableau, we get an SKB: the original normal form tableau did not have superfluous rows, so the quantity of the rows in the resulting tableau will be minimal; furthermore, as the normal form tableau does not contain any false Ws, the resulting tableau will have the maximal number of *es*.

In the other direction, if we split all rows of an SKB into one-L rows, there can be no superfluous rows in the result (otherwise the L corresponding to a superfluous row could have been replaced with an *e*, contrary to the definition of an SKB which must have as many *es* as possible); as for false Ws, there can be none in the SKB tableau itself, and after all row splittings are applied, no new false Ws can arise (if a false W could arise in one of the resulting one-L rows, then the same W would have been false even before splitting).

What remains is to show that there can be no two SKBs in the same equivalence class. Suppose towards contradiction there are two SKBs  $S_1$  and  $S_2$ . They both normalize to the same normal form tableau by the procedure above. From the definition of SKB, only row splittings are required. Pick an arbitrary set of rows  $r_1, \dots, r_n$  with Ws in the same constraints from the resulting normal form tableau. If  $S_1$  and  $S_2$  each have only one row splitting into this same set, that must be the same row. If  $S_1$  and  $S_2$  have more than one row splitting into this set, we can actually merge those rows into just one, resulting in a smaller equivalent tableau  $S_3$ , contrary to the assumption of  $S_1$  and  $S_2$ 's minimality. Thus either  $S_1 = S_2$ , or they are not minimal possible size in their equivalence class. Therefore there is only one SKB per class.  $\square$

Theorem (18) essentially means that all useful results about normal forms may be transferred to SKBs. For instance, the equivalence test in Theorem (16) may be replaced by an equivalence test comparing SKBs derived using Brasoveanu and Prince's Fusional Reduction algorithm. With 18 in hand, we may employ Brasoveanu and Prince's SKBs as representatives of their equivalence classes instead of our normal forms. Normal forms are often more convenient in complex proofs, because the relations between constraints in them are maximally untangled; but SKBs are more useful when it becomes convenient to have smaller-sized representatives.

5

## CONCLUSION

We defined a normal form for OT tableaux, and showed that there is exactly one normal form in each equivalence class of OT tableaux. Moreover, we have demonstrated that each OT tableau can be computably normalized by a sequence of five pairs of previously known equivalence-preserving transformations in Operations (3)–(7). The computational cost of normalization is only polynomial in the number of rows  $m$  and constraints  $n$ , thanks to the use of the efficient RCD-based algorithm for entailment checking proposed by Prince (2002, Section 5).

Those results provide us with a handle on equivalence classes of OT tableaux: using them, we may reason about tableaux without any loss of generality while only considering normal forms. The examples in Section 4, including Theorem (18) stating that Brasoveanu and Prince's Skeletal Bases are unique in their equivalence classes, illustrate how to capitalize on the presented OT normal form theorems.

## REFERENCES

- Adrian BRASOVEANU and Alan PRINCE (2005), Ranking and Necessity, Part I, Rutgers Optimality Archive 794.
- Adrian BRASOVEANU and Alan PRINCE (2011), Ranking and Necessity: the Fusional Reduction Algorithm, *Natural Language and Linguistic Theory*, 29(1):3–70, revised version of Brasoveanu and Prince (2005).

- Bruce HAYES (1997), Four Rules of Inference for Ranking Argumentation, ms., UCLA. <http://www.linguistics.ucla.edu/people/hayes/otsoft/argument.pdf>.
- Renè KAGER (1999), *Optimality Theory*, Cambridge University Press, Cambridge.
- Giorgio MAGRI (2009), *A Theory of Individual-Level Predicates Based on Blind Mandatory Implicatures. Constraint Promotion for Optimality Theory*, Ph.D. thesis, MIT. <http://dspace.mit.edu/handle/1721.1/55182>.
- Alan PRINCE (2000), Comparative tableaux, Rutgers Optimality Archive 376.
- Alan PRINCE (2002), Entailed Ranking Arguments, Rutgers Optimality Archive 500. <http://roa.rutgers.edu/article/view/510>.
- Alan PRINCE (2006), No more than Necessary: beyond the Four Rules, and a bug report, Rutgers Optimality Archive 882. <http://roa.rutgers.edu/article/view/905>.
- Alan PRINCE and Paul SMOLENSKY (1993), Optimality Theory: Constraint Interaction in Generative Grammar, Rutgers Optimality Archive 537. <http://roa.rutgers.edu/article/view/547>.
- Alan PRINCE and Paul SMOLENSKY (2004), *Optimality Theory: Constraint Interaction in Generative Grammar*, Blackwell, Oxford.
- Igor YANOVICH (2011), On sets of OT rankings, Rutgers Optimality Archive 1149. <http://roa.rutgers.edu/article/view/1203>.

*This work is licensed under the Creative Commons Attribution 3.0 Unported License.*  
<http://creativecommons.org/licenses/by/3.0/>

