

## CHAPTER 2

### A RANK-ORDERING MODEL OF EVAL

An Optimality Theoretic grammar (Prince and Smolensky, 1993) consists of three major components, namely a generative function (GEN), a universal set of constraints (CON), and an evaluative function (EVAL). GEN is a universal function that generates a candidate set for any given input. CON contains all of the constraints that make up Universal Grammar. EVAL uses the constraints in CON to compare the candidates generated by GEN with each other in order to determine the output associated with an input. EVAL can therefore be considered as the center of an OT grammar – this is where, given the candidate set and the constraints, the output of the grammar is determined.

In this chapter of the dissertation I investigate the formal properties of EVAL in more detail. Specifically, I develop a set theoretic model of EVAL. This chapter serves two purposes: (i) In the rest of this dissertation I will argue that EVAL works somewhat differently from what has traditionally been assumed in the OT literature. I will make two claims about EVAL. First, rather than just distinguishing between the best candidate and the mass of losers, EVAL imposes a harmonic rank-ordering on the full candidate set. Secondly, EVAL can compare any set of candidates, irrespective of whether they are related to each via a shared input or not. (For more on these two claims, refer to Chapter 1.) This chapter shows that this alternative view of EVAL is entirely compatible with the architecture of a classic OT grammar and does not require any formal changes to the architecture of the grammar. (ii) But this chapter also serves the more general purpose of providing a mathematical model of EVAL. Once EVAL is formulated as a mathematical

object, many of the assumptions about an OT grammar that are implicitly part of OT literature, can be explicitly stated and formally proved.

The basic approach in this chapter is that of “explication” which Carnap defines as “transforming a given more or less inexact concept into an exact one, or rather, replacing the first by the second” (Carnap, 1962:3). Carnap calls the inexact concept that explication sets out to formalize the “explicandum”, and the result of the explication process the “explicatum”. The goal of this chapter is therefore not to propose a new theory of grammar, but rather to express in explicit, formal terms what is generally accepted about OT. Since explication is basically a descriptive activity, it is quite hard to decide whether the explicatum is “correct” or not (Carnap, 1962:4).<sup>1</sup> The correctness of the explicatum should be measured by how well it resembles the explicandum. However, since the explicandum is by definition not an exact, precise concept, it is difficult (if not in principle impossible) to determine whether the explicatum exactly fits the explicandum. Even so, we should be confident that the explicatum at least agrees with our basic intuitions about the explicandum. Throughout the discussion, I will therefore point out how the model of EVAL that I am developing resembles what is standardly accepted about an OT grammar. One of the most characteristic features of an OT grammar is the so-called “strictness of strict domination” principle (McCarthy, 2002b:4, Prince and Smolensky, 1993:78, 1997:1604). I will therefore in particular show that the model of EVAL developed here abides by this principle (see §3.2.1).

---

<sup>1</sup> See also Kornai (1995:xix-xxi) for a discussion of this same problem specifically with regard to formalizing linguistic theories.

The chapter is structured as follows: In §1 I give a short review of how the rank-ordering model of EVAL is different from what is usually assumed about EVAL in OT literature. I also discuss previous mathematical formalizations of EVAL, and show that they are not consistent with a rank-ordering model of EVAL. Section §2 gives a characterization of constraints as functions from the candidate set into  $\mathbb{U}$ , and then shows how the candidate set can be ordered with respect to individual constraints. In §3 I show how the orderings associated with individual constraints are combined into one single ordering for the whole grammar. The chapter ends in two appendices. Appendix A contains a list of all the definitions used in the chapter, and Appendix B a list of the theorems and lemmas formulated.

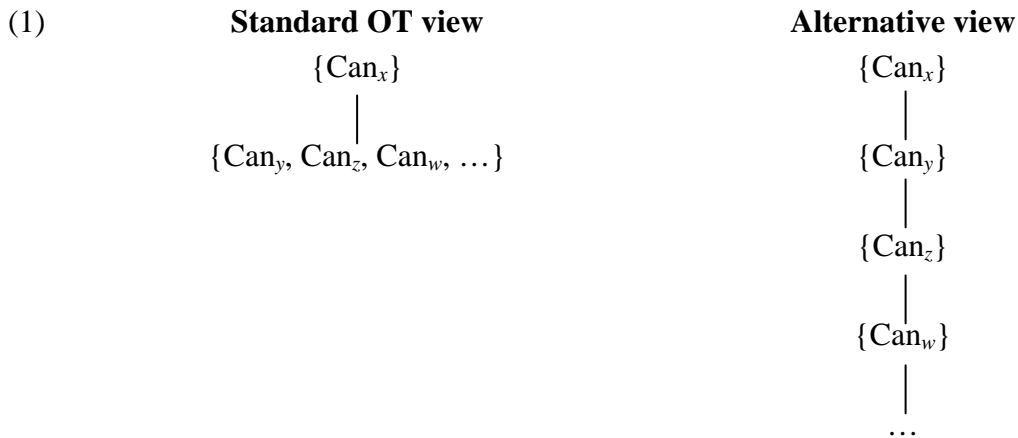
This chapter is somewhat independent from the rest of the dissertation. Since it contains many results about an OT grammar that are not directly relevant to the rest of the dissertation, it can be read as a self-contained unit without reading the rest of the dissertation. Similarly, the rest of the dissertation can also be read without reading this chapter. Any of the issues discussed in this chapter that are relevant elsewhere in the dissertation, are also discussed where they are relevant. However, this chapter contains the only comprehensive formal treatment of the theoretical assumptions made in this dissertation. Reading this chapter will enhance the overall understanding of the theoretical claims. Readers who are not mathematically inclined may either skip this chapter completely, or jump ahead in this chapter to §4. In section §4 I provide a brief summary the chapter, and point out which of the results of this chapter will be relevant in the rest of the dissertation.

## 1. A rank-ordering model of EVAL

Classic OT is a theory of winners. It makes only one distinction in the candidate set, between the winning candidate and the mass of losers. Once a candidate has been eliminated from the race, it is demoted to the set of non-optimals or losers. And once in this set of losers, all information supplied about the candidate by the constraints becomes irrelevant. All losers are treated alike – as members of one large amorphous set.

This standard view of an OT grammar is held in spite of the fact that the theory can make finer grained distinctions in the candidate set. If we remove the optimal candidate from a candidate set and consider only the set of losers, then there will again be a candidate that is better than all the rest. This best candidate amongst the losers can then be removed, and we can repeat the comparison again to find the best candidate in the remaining smaller set of losers. In fact, this process can be repeated for as long as there are still candidates left, and we can therefore rank-order the full candidate set in this way.

These two views about the output of an OT grammar are represented graphically in (1). Candidates appearing higher are more harmonic relative to the constraint ranking. The “alternative view” is the view that I am assuming in this dissertation.



Since the information about the relationships between losers is considered irrelevant in classic OT, previous mathematical models of EVAL were formulated to ignore this information. These models can be divided into two groups: (i) models that are formulated such the information about the relationships between the non-winners is not generated at all; (ii) models that generate this information but ignore it. I will discuss one example of each kind here.

As an example of the first kind, consider the model developed by Samek-Lodovici and Prince (1999, Prince, 2002). They view constraints as functions from sets of candidates to sets of candidates. A constraint takes as input a set of candidates and returns as output the subset consisting of those candidates that perform best on the particular constraint. All other candidates are demoted to the set of losers, irrespective of how they are related to each other. The set of losers might contain candidates that violate the particular constraint only once and also candidates that violate the constraint many times. No distinction is made between losers. EVAL is then simply a composition of the different constraint functions in the order in which they are ranked. For instance, if  $K$  is the candidate set generated by GEN, and  $\|C_1 \circ C_2 \circ \dots \circ C_n\|$  the constraint hierarchy, then EVAL will have the form  $(C_n \text{ B } \dots \text{ B } C_2 \text{ B } C_1)(K) = C_n(\dots C_2(C_1(K)))$ . EVAL then returns only the single best (set of) candidate(s) as output. The complement in  $K$  of the set returned by EVAL is the set of losers. But no information is available about how the candidates in this set of losers are related to each other. Other models that were developed along similar lines are those of Eisner (1997a, 1997b, 1999), Hammond (1997) and Karttunen (1998).

Moreton (1999) has a different conceptualization of EVAL, and his model serves as an example of the second type of model of EVAL. In Moreton's model the information about the relationships between losers is generated but ignored. The end result is that his model still makes only a two-level distinction in the candidate set. For Moreton, constraints are functions from the set of candidates into  $\mathbb{N}$ . A constraint takes a single candidate as input, and then maps the candidate onto the natural number corresponding to the number of times that the candidate violates the particular constraint. The fact that constraints are functions on the candidate set implies that every constraint applies to every candidate. This is the crucial difference between Moreton's model and the models discussed above. In the other models, every constraint prunes the candidate set down so that later constraints may not get the opportunity of applying to the full candidate set. Since all constraints evaluate all candidates in Moreton's model, the losers can in principle also be compared with each other. However, as shown below, Moreton conceptualizes EVAL in such a way that this does not happen.

Moreton defines for each candidate a *score vector*. The score vector of a candidate consists of the number of violations afforded the candidate by each of the constraints, ordered according to the ranking between the constraints. For instance, consider a constraint hierarchy  $\|C_1 \circ C_2 \circ C_3\|$ , and a candidate  $k$  that violates  $C_1$  once,  $C_2$  three times, and  $C_3$  twice, i.e.  $C_1(k) = 1$ ,  $C_2(k) = 3$ , and  $C_3(k) = 2$ . The score vector associated with  $k$  is then  $v_k = \langle C_1(k), C_2(k), C_3(k) \rangle = \langle 1, 3, 2 \rangle$ . Every candidate has such a score vector associated with it. Score vectors are compared as stated in (2).

(2) **Comparing score vectors in Moreton's model**

Let  $v = \langle s_1, s_2, \dots, s_n \rangle$  and  $v' = \langle s'_1, s'_2, \dots, s'_n \rangle$  be score vectors.

We say that  $v < v'$  iff  $\exists j \leq n$  such that:

- (i)  $\forall i < j: s_i = s'_i$ , and
- (ii)  $s_j < s'_j$

The score vector of some candidate  $k_1$  precedes the score vector of some other candidate  $k_2$  if the highest ranked constraint that judges the two candidates differently favors  $k_1$  over  $k_2$ . Moreton then defines the output of the grammar (of EVAL) for some input as that candidate whose score vector precedes the score vectors of all other candidates. Although the information about the relationships between the other candidates is generated, this information is ignored in the final output of the grammar where only the best candidate is distinguished from the mass of losers. De Lacy has a similar characterization of constraints (de Lacy, 2002:30).

These earlier models are therefore not compatible with a rank-ordering model of EVAL. Models like that of Samek-Lodovici and Prince (1999) are in principle incompatible since they do not even generate the information that would be required to rank-order the full candidate set. Models such as those of Moreton (1999) generate this information and are therefore in principle compatible with a rank-ordering model of EVAL. However, these models are formulated in such a way that this information is ignored.

In the rest of this chapter I will develop a model of EVAL that generates the information about the relationships between the losers, and also uses this information

explicitly to impose a rank-ordering on the full candidate set. Although Moreton's model can in principle be extended to do this, I choose to formulate a model that is different from Moreton's. In Moreton's model, comparison between candidates is done in terms of score vectors. In the model that I develop comparison is done in terms of individual constraints – every constraint imposes a rank-ordering on the candidate set, and the orderings associated with different constraints are then combined to yield a final ordering for the full grammar. EVAL orders the candidate set in two stages, first in terms of individual constraints and then by combining the orderings associated with individual constraints. The next two sections discuss each of these two stages in the ordering process.

## **2. EVAL and the ordering associated with individual constraints**

The comparison that EVAL makes is based on the violations that the constraints assign to each candidate. Before we can consider the properties of EVAL, it is therefore necessary to have a clear idea of what constraints are. In this section I will first characterize constraints, and only then show how EVAL orders the candidate set with respect to individual constraints.

### **2.1 Characterization of constraints**

A constraint considers each of the candidates generated by GEN separately, and evaluates that candidate according to some substantive requirement. A constraint assigns a violation mark to a candidate for every instance of non-compliance of the candidate with the specific substantive requirement of that constraint. A constraint therefore sets up a



relation between a candidate and a number of violations. We can regard the domain of this relation as the set of all candidates, and its range as a subset of  $\mathbb{N}$ , the natural numbers. Constraints can then be characterized as in (3).

(3) **Constraints as relations between the candidate set and  $\mathbb{N}$**

Let CON be the universal set of constraints, and  $K$  the set of candidates to be evaluated. Then,  $\forall C \in \text{CON}$ :

$C: K \rightarrow \mathbb{N}$  such that  $\forall k \in K, C(k) =$  number of violations of  $k$  in terms of  $C$

This characterization of constraints is basically the same as that assumed by Moreton (1999).<sup>2</sup> However, it is significantly different from the view taken by Samek-Lodovici and Prince (Prince, 2002, Samek-Lodovici and Prince, 1999). For them, constraints take as argument not individual candidates, but sets of candidates. Also, a constraint does not return a natural number as its value, but a subset of the candidate set that it took as argument. (See discussion above in §1.)

---

<sup>2</sup> It is also in principle identical to the way that de Lacy (2002:30) defines constraints. For de Lacy a constraint is a relation that takes as input a candidate, and returns not a natural number but a set of violation marks. Comparison between candidates is then done by comparing the cardinality of the sets of violation marks of each candidate. However, since a repeated identical element in a set does not change the set (i.e.  $\{*\} = \{*, *, *, \dots\}$ ), de Lacy has to introduce a method to make multiple violation marks distinct. He needs a way in which a set with  $n$  violation marks will have a cardinality of  $n$ . This is his solution: “To avoid this problem, take a ‘violation mark’ to be any element from a denumerably infinite set of discrete elements (e.g. the natural numbers). Thus, a set of three violation marks is  $\{1,2,3\}$ , with a cardinality of 3.” (2002:30). A constraint is then a relation that maps each candidate onto a set with cardinality equal to the number of times that the candidate violates the constraint.

This is in principle identical to the characterization of constraints that I give in (3) above. The natural numbers can be reconstructed in set theoretic terms such that each natural number is simply a set with cardinality equal to the specific natural number, i.e. the natural number  $n$  is a set with cardinality  $n$  (Enderton, 1977:66-89). When we think about the natural numbers in set theoretic terms, then the way in which constraints are characterized above in (3) can be seen as relations that map each candidate onto a set with cardinality equal to the number of times that the candidate violates the constraint. De Lacy’s characterization of constraints is therefore only superficially different from the view that I take.

Based on some generally accepted properties of an OT grammar we can show that constraints are functions. The definition of a function in (4) comes from Partee *et al.* (1993:30).

(4) **Def. 1: Functions**

A relation  $R$  from  $A$  to  $B$  is a function iff:

- (a) the domain of  $R = A$  (i.e. every member of  $A$  is mapped onto some member of  $B$ ), and
- (b) each element in  $A$  is mapped onto just one element in  $B$  ( $R$  is single valued).

(5) **Theorem 1: Constraints as functions**

All constraints are functions.

It is not possible to prove Theorem 1. The truth of this Theorem does not follow from some inherent property of what it means to be a constraint, but from the way in which constraints are conventionally formulated in OT. The discussion in this paragraph is therefore not a proof, but only illustrative in nature. In order for (4a) to hold of constraints, it is necessary that every constraint assign some value to every candidate. This follows from the assumption that a constraint applies even to candidates that do not violate the constraint – it assigns the natural number zero to these candidates. In order for (4b) to hold, a constraint should assign a unique value to each candidate. This is obviously true. A candidate violates a constraint a fixed number of times, and this number of violations is the only value that the constraint can assign to a candidate.

Throughout this chapter I will use an example to illustrate the concepts that I discuss. In this example I will assume a grammar with only three constraints, i.e.  $\text{CON} = \{C_1, C_2, C_3\}$  and only five candidates, i.e.  $K = \{c_1, c_2, c_3, c_4, c_5\}$ . I will also assume a specific ranking between the constraints, namely  $\|C_1 \circ C_2 \circ C_3\|$ . The tableau in (6) shows how each of the five candidates is evaluated by the three constraints in this example.

(6)  $\{c_1, c_2, c_3, c_4, c_5\}$  evaluated by  $\|C_1 \circ C_2 \circ C_3\|$

	$C_1$	$C_2$	$C_3$
$c_1$	*	**	
$c_2$	**	*	*
$c_3$		*	**
$c_4$	*	**	
$c_5$	**		**

In (3) constraints were characterized as relations between  $K$  and  $\dot{U}$ . This can now be illustrated: Since  $c_1$  violates  $C_1$  once, it means that  $C_1$  will map  $c_1$  onto the value 1, i.e.  $C_1(c_1) = 1$ . Similarly,  $C_1(c_2) = 2$ ,  $C_1(c_3) = 0$ , etc. We can do the same for all three constraints and all five candidates. We can also represent each constraint as a set of ordered pairs  $\langle x, y \rangle$  where  $x$  is a candidate and  $y$  the value onto which the constraint maps  $x$ . In (7) I show these sets of ordered pairs for every constraint.

(7) Constraints as relations between  $K$  and  $\dot{U}$

$$C_1 = \{\langle c_1, 1 \rangle, \langle c_2, 2 \rangle, \langle c_3, 0 \rangle, \langle c_4, 1 \rangle, \langle c_5, 2 \rangle\}$$

$$C_2 = \{\langle c_1, 2 \rangle, \langle c_2, 1 \rangle, \langle c_3, 1 \rangle, \langle c_4, 2 \rangle, \langle c_5, 0 \rangle\}$$

$$C_3 = \{\langle c_1, 0 \rangle, \langle c_2, 1 \rangle, \langle c_3, 2 \rangle, \langle c_4, 0 \rangle, \langle c_5, 2 \rangle\}$$

Theorem 1 (5) stated that constraints are functions. It is clear that the relations in (7) are functions. First, each of the five candidates in  $K$  is represented by an ordered pair in each of these sets. Secondly, every candidate is mapped onto only one value.

## 2.2 Ordering the candidates with respect to individual constraints

A constraint sees candidates only in terms of their violations. Two candidates that earn the same number of violations in terms of some constraint are therefore indistinguishable from each other as far as that constraint is concerned.<sup>3</sup> This means that the two candidates [a.ta] and [pu.i.ma], although they are clearly distinct, cannot be distinguished in terms of the constraint ONSET (every syllable must have an onset). Both of these candidates contain one onsetless syllable, and they are therefore both mapped onto the same value by the constraint ONSET, i.e.  $\text{ONSET}([a.ta]) = \text{ONSET}([pu.i.ma]) = 1$ . Although these two candidates are distinct, they share with each other all ordering relationships in terms of the constraint ONSET to the rest of the candidate set.

When we consider the ordering that EVAL imposes on the candidate set with reference to a specific constraint, it is therefore not necessary to consider an ordering that refers to every candidate individually. Rather, the ordering can be viewed as an ordering defined on sets of candidates, specifically on sets of candidates that share the same number of violations. This leads to a significant simplification by reducing the number of discrete elements that need to be compared.

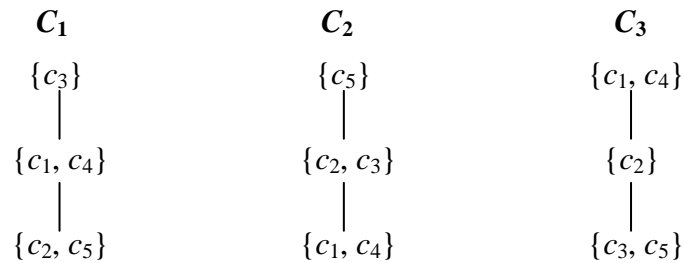
This can also be illustrated with the example introduced from (6) above. In this example candidates  $c_2$  and  $c_5$  are clearly distinct – since they violate  $C_2$  and  $C_3$  to

---

<sup>3</sup> See Samek-Lodovici and Prince (1999) about such “grammatically indistinct” candidates. Sections §2.2.1 and §3.2.5 below contain more discussion of grammatical indistinctness in OT.

different degrees. Even so, in terms of  $C_1$  they are indistinguishable – they both violate  $C_1$  twice. Candidates  $c_2$  and  $c_5$  will therefore occupy the same slot in the ordering that EVAL imposes on the candidate set in terms of  $C_1$ . The same is true of candidates  $c_1$  and  $c_4$ . In the same manner we can also establish the ordering that each of  $C_2$  and  $C_3$  will impose on the candidate set. In (8) I represent the orderings associated with each of these constraints graphically. Candidates (more accurately, sets of candidates) that appear higher in this graphic representation are rated better by the particular constraint.

(8) **Orderings imposed by EVAL on candidate set in terms of  $C_1$ ,  $C_2$  and  $C_3$**



Since this ordering is clearly on sets of candidates and not on candidates, the first thing we need to do is to gather individual candidates into the sets on which the ordering is defined. The candidates in each of these sets are those candidates with the same number of violations in terms of the specific constraint. We can therefore define a relation that will express what the candidates in each of these sets have in common.

(9) **Def. 2: The relation  $\approx_C$  on  $K$**

Let  $K$  be the candidate set to be evaluated by EVAL, and CON the set of constraints.

Then, for all  $k_1, k_2 \in K$ , and for all  $C \in \text{CON}$ , let:

$$k_1 \approx_C k_2 \quad \text{iff} \quad C(k_1) = C(k_2).$$

Consider  $C_1$  in the example discussed above. Candidates  $c_1$  and  $c_4$  both earn 1 violation in terms of  $C_1$ , i.e.  $C_1(c_1) = C_1(c_4) = 1$ . From this it follows that  $c_1$  and  $c_4$  stand in the  $\approx_{C_1}$ -relation to each other, or  $c_1 \approx_{C_1} c_4$ . Since  $=$  is reflexive, we of course also have  $c_4 \approx_{C_1} c_1$ . And since  $=$  is also symmetric we also have  $c_i \approx_{C_1} c_i$  for each candidate  $c_i$ . In the same way the  $\approx_{C_i}$ -relation can be determined for each constraint. In (10) I show these relationships for each of the constraints.

(10)  **$\approx_C$ -relations for each constraint**

$$C_1(c_i) = 0: \quad c_3 \approx_{C_1} c_3$$

$$C_1(c_i) = 1: \quad c_1 \approx_{C_1} c_4, c_4 \approx_{C_1} c_1, \quad c_1 \approx_{C_1} c_1, c_4 \approx_{C_1} c_4$$

$$C_1(c_i) = 2: \quad c_2 \approx_{C_1} c_5, c_5 \approx_{C_1} c_2, \quad c_2 \approx_{C_1} c_2, c_5 \approx_{C_1} c_5$$

$$C_2(c_i) = 0: \quad c_5 \approx_{C_2} c_5$$

$$C_2(c_i) = 1: \quad c_2 \approx_{C_2} c_3, c_3 \approx_{C_2} c_2, \quad c_2 \approx_{C_2} c_2, c_3 \approx_{C_2} c_3$$

$$C_2(c_i) = 2: \quad c_1 \approx_{C_2} c_4, c_4 \approx_{C_2} c_1, \quad c_1 \approx_{C_2} c_1, c_4 \approx_{C_2} c_4$$

$$C_3(c_i) = 0: \quad c_1 \approx_{C_3} c_4, c_4 \approx_{C_3} c_1, \quad c_1 \approx_{C_3} c_1, c_4 \approx_{C_3} c_4$$

$$C_3(c_i) = 1: \quad c_2 \approx_{C_3} c_2,$$

$$C_3(c_i) = 2: \quad c_3 \approx_{C_3} c_5, c_5 \approx_{C_3} c_3, \quad c_3 \approx_{C_3} c_3, c_5 \approx_{C_3} c_5$$

We can show that the relation  $\approx_C$  is an equivalence relation, which basically means that two elements that stand in the  $\approx_C$ -relation to each other are indistinguishable from each other in terms of this relation. In (11) I state the requirements that must be met for a relation to be an equivalence relation (Enderton, 1977:56), and in (12) I then show that  $\approx_C$  is indeed an equivalence relation.

(11) **Def. 3: An equivalence relation**

A binary relation  $R$  on some set is an equivalence relation on that set iff  $R$  is

(i) reflexive, (ii) symmetric, and (iii) transitive.

(12) **Theorem 2:  $\approx_C$  as an equivalence relation**

For all  $C \in \text{CON}$ ,  $\approx_C$  is an equivalence relation on  $K$ .

This is obviously true.  $\approx_C$  is by definition a binary relation. Also,  $\approx_C$  is defined in terms of the relation  $=$  on  $\mathbb{U}$ , and  $=$  is reflexive, symmetric and transitive on  $\mathbb{U}$ . The relation  $\approx_C$  therefore inherits these properties from  $=$ .

Since  $\approx_C$  is an equivalence relation, we can use  $\approx_C$  to define equivalence classes (Enderton, 1977:57) on the candidate set. An equivalence class in terms some equivalence relation  $R$  is a set containing all the forms that stand in the relation  $R$  to each other. In terms of the relation  $\approx_C$  there will therefore be an equivalence class containing the candidates that earn zero violations in terms of  $C$ , a class containing the candidates that earn one violation in terms of  $C$ , a class containing the candidates that earn two violations in terms of  $C$ , etc.

(13) **Def. 4: Equivalence classes on  $K$  in terms of  $\approx_C$**

For all  $k_1 \in K$ ,  $f_{k_1, C} := \{k_2 \in K \mid k_1 \approx_C k_2\}$

Every candidate  $k_1$  will therefore be grouped together into a set (an equivalence class) with all the other candidates that receive the same number of violations as  $k_1$  in

terms of  $C$ . In (14) I show the equivalence classes that are defined on  $K$  in terms of each of the  $\approx_C$ -relations in our example grammar from (6).

(14) **Equivalence classes on the candidate set in terms of  $\approx_C$ -relations**

$$\begin{array}{llll}
 C_1(c_i) = 0: & fc_{3,C1} & = & \{c_3\} \\
 C_1(c_i) = 1: & fc_{1,C1} = fc_{4,C1} & = & \{c_1, c_4\} \\
 C_1(c_i) = 2: & fc_{2,C1} = fc_{5,C1} & = & \{c_2, c_5\} \\
 \\ 
 C_2(c_i) = 0: & fc_{5,C2} & = & \{c_5\} \\
 C_2(c_i) = 1: & fc_{2,C2} = fc_{3,C2} & = & \{c_2, c_3\} \\
 C_2(c_i) = 2: & fc_{1,C2} = fc_{4,C2} & = & \{c_1, c_4\} \\
 \\ 
 C_3(c_i) = 0: & fc_{1,C3} = fc_{4,C3} & = & \{c_1, c_4\} \\
 C_3(c_i) = 1: & fc_{2,C3} & = & \{c_2\} \\
 C_3(c_i) = 2: & fc_{3,C3} = fc_{5,C3} & = & \{c_3, c_5\}
 \end{array}$$

The ordering that EVAL imposes on the candidate set will be defined in terms of these equivalence classes – i.e. EVAL does not order candidates directly, but rather orders equivalence classes of candidates. However, orderings are defined on elements of a set, and at this moment these equivalence classes do not form a set. The next step we need to accomplish is therefore to collect all of the equivalence classes into one set on which the ordering can then be defined. The set that has as its members all of the equivalence classes on some set  $A$  in terms of some equivalence relation  $R$ , is known as the quotient set of  $A$  modulo  $R$  (Enderton, 1977:58). We can define such a quotient set on the candidate set  $K$  modulo the equivalence relation  $\approx_C$ . The definition is given in



(15), and the quotient sets associated with each of the constraints in our example are given in (16).

(15) **Def. 5: Quotient set on  $K$  modulo  $\approx_C$**

$$K/C := \{fk_{,C} \mid k \in K\}$$

(16) **Quotient sets on  $K$  modulo  $\approx_C$  for each constraint**

$$K/C_1 = \{\{c_3\}, \{c_1, c_4\}, \{c_2, c_5\}\}$$

$$K/C_2 = \{\{c_5\}, \{c_2, c_3\}, \{c_1, c_4\}\}$$

$$K/C_3 = \{\{c_2\}, \{c_1, c_4\}, \{c_3, c_5\}\}$$

We are now finally in a position to define the ordering that EVAL imposes on (the quotient set on) the candidate set. In this ordering the equivalence class with the candidates that receive the smallest number of violations in terms of  $C$  will occupy the first position (will be the minimum, will precede all other equivalence classes), next will be the equivalence class with candidates that receive the second smallest number of violations in terms of  $C$ , etc.

(17) **Def. 6: The ordering relation  $\leq_C$  on the set  $K/C$**

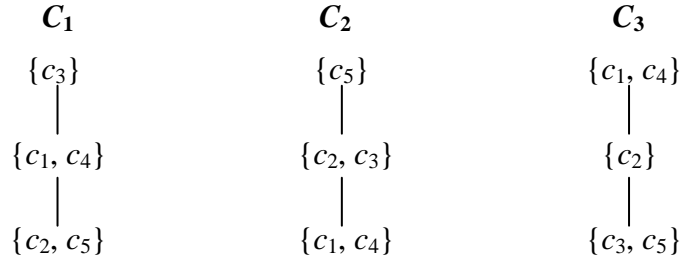
For all  $C \in \text{CON}$  and all  $fk_{1,C}, fk_{2,C} \in K/C$ :

$$fk_{1,C} \leq_C fk_{2,C} \text{ iff } C(k_1) \leq C(k_2).$$

Equivalence class  $fk_{1,C}$  “precedes” or “is better than” equivalence class  $fk_{2,C}$  if the candidates belonging to  $fk_{1,C}$  receives fewer violations in terms of  $C$  than the candidates belonging to  $fk_{2,C}$ . The ordering  $\leq_C$  that each of the constraints in our example imposes

on the candidate set is shown in (18). Note that this is exactly the same as the orderings shown in (8) above.

(18) **The  $\leq_{C_i}$ -ordering that EVAL imposes on each quotient set  $K/C_i$**



The relation  $\leq_C$  imposes an ordering on the quotient set  $K/C$  resulting in the ordered set  $\langle K/C, \leq_C \rangle$  for each constraint. This set is the output of EVAL with respect to constraint  $C$ . Since the output of EVAL with respect constraints is defined in terms of the set  $K/C$  and ordering  $\leq_C$ , the characteristics of this set and ordering are important. In the next few sub-sections I discuss those characteristics of this set and ordering that are most directly relevant to our understanding of what a grammar is. First, I discuss the relation of individual candidates to the ordered set  $\langle K/C, \leq_C \rangle$  (§2.2.1). Then I show that the ordering defined by  $\leq_C$  is a chain (§2.2.2), and that this chain is guaranteed to always have a minimum (§2.2.3). Finally, I show that the set  $K/C$  is a partition on  $K$  (§2.2.4). The relevance of each of these results is discussed in the respective sections.

### 2.2.1 Individual candidates in the set $\langle K/C, \leq_C \rangle$

The ordering  $\leq_C$  was defined in (17) above in terms of equivalence classes of candidates and not in terms of candidates. However, in actual practice we are usually interested in the relationship between candidates, and not between equivalence classes of candidates. In this section I will show that it is trivial matter to move from the ordering relationship

between equivalence classes of candidates to the relationship between individual candidates. It therefore does not matter whether we think of the ordering that EVAL imposes as an ordering on equivalence classes of candidates, or as an ordering on the individual candidates. I will first define an ordering on individual candidates in terms some constraint (this is the ordering that we are interested in when we actually do grammatical analyses), and then I will show that it is a trivial matter to move from the ordering on equivalence classes of candidates to this ordering on individual candidates.

(19) **Def. 7: The ordering relation  $\leq_{C'}$  on the set  $K$**

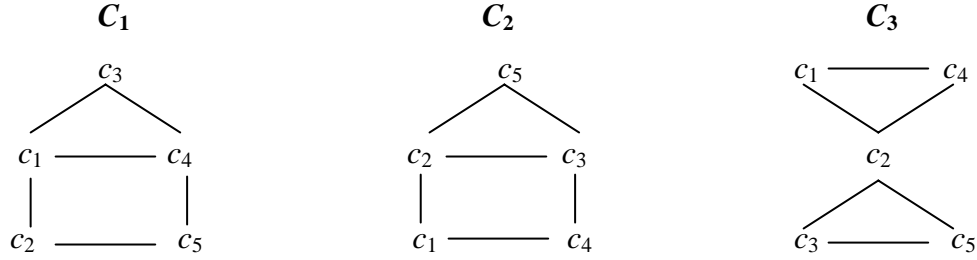
For all  $C \in \text{CON}$  and all  $k_1, k_2 \in K$ :

$$k_1 \leq_{C'} k_2 \text{ iff } C(k_1) \leq C(k_2)$$

The ordering  $\leq_{C'}$  is intuitive –  $k_1$  “precedes” or “is better than”  $k_2$  if and only if  $k_1$  receives fewer violations in terms  $C$  than  $k_2$ . The ordering  $\leq_{C'}$  therefore defines a ordering on the candidate set  $K$ , resulting in the ordered set  $\langle K, \leq_{C'} \rangle$ . In (20) I give a graphic representation of the  $\leq_{C'}$ -orderings associated with each of the three constraints in our example. In this representation a candidate that appears higher precedes a candidate that appears lower – i.e. if  $k_1$  appears higher than  $k_2$ , then  $k_1 \leq_{C'} k_2$ . If two candidates appear next to each other, then they are equal in terms of the ordering – i.e. if  $k_1$  appears next to  $k_2$ , then  $k_1 =_{C'} k_2$ .

There is a natural relationship between the ordering  $\leq_{C'}$  on candidates and the ordering  $\leq_C$  on the equivalence classes –  $k_1 \leq_{C'} k_2$  is only possible if  $fk_{1,C} \leq_C fk_{2,C}$  and *vice versa*. That this is the case is obvious – these two orderings are defined by the same condition, i.e. if and only if  $C(k_1) \leq C(k_2)$ , then  $k_1 \leq_{C'} k_2$  and  $fk_{1,C} \leq_C fk_{2,C}$ .

(20) **The  $\leq_c$ -ordering that EVAL imposes on the candidate set for each constraint**



Because of this natural relationship between these two orderings it is a trivial matter to move from the ordered set  $\langle K/C, \leq_c \rangle$  to the ordered set  $\langle K, \leq_{c'} \rangle$ . Suppose that we have only the ordered set  $\langle K/C, \leq_c \rangle$ , i.e. the set that contains not candidates but equivalence classes of candidates. Suppose further that we want to know how two candidates  $k_1$  and  $k_2$  are related to each other in terms of the ordering on individual candidates  $\leq_{c'}$ . All we need to do is to find the equivalence classes  $fk_{1,c}$  and  $fk_{2,c}$ . If  $fk_{1,c} <_c fk_{2,c}$ , then also  $k_1 <_{c'} k_2$ . If  $fk_{1,c} =_c fk_{2,c}$ , then also  $k_1 =_{c'} k_2$ . If  $fk_{1,c} >_c fk_{2,c}$ , then also  $k_1 >_{c'} k_2$ . This can easily be proven formally. What we need is to show that there exists an order-embedding mapping from  $\langle K/C, \leq_c \rangle$  to  $\langle K, \leq_{c'} \rangle$ . In (21) I first define what an order-embedding mapping is. In (22) I then define a mapping  $\psi$  from  $\langle K/C, \leq_c \rangle$  to  $\langle K, \leq_{c'} \rangle$ . In (23) I show the result of applying  $\psi$  to the ordered sets  $\langle K/C, \leq_c \rangle$  associated with each of the constraints in our example. Following that, I show that this mapping  $\psi$  is an order-embedding. The definition of an order-embedding is from Davey and Priestly (1990:10).

(21) **Def. 8: An order-embedding**

Let  $P$  and  $Q$  be ordered sets. A map  $\varphi: P \rightarrow Q$  is said to be an order-embedding if  $x \leq y$  in  $P$  iff  $\varphi(x) \leq \varphi(y)$  in  $Q$ .

(22) **Def. 9: A mapping from  $\langle K/C, \leq_C \rangle$  to  $\langle K, \leq_{C'} \rangle$**

$\psi: \langle K/C, \leq_C \rangle \rightarrow \langle K, \leq_{C'} \rangle$  such that:

For all  $fk_{x', C} \in K/C$  and for all  $k_y \in fk_{x', C}$ ,  $\psi(fk_{x', C}) = k_y$ .<sup>4</sup>

(23) **Mapping equivalence classes onto their members**

a.  **$\psi$  applied to  $\langle K/C_1, \leq_{C_1} \rangle$ :**

$$\psi(\{c_3\}) = c_3$$

$$\psi(\{c_1, c_4\}) = c_1 \text{ and } c_4$$

$$\psi(\{c_2, c_5\}) = c_2 \text{ and } c_5$$

b.  **$\psi$  applied to  $\langle K/C_2, \leq_{C_2} \rangle$ :**

$$\psi(\{c_5\}) = c_5$$

$$\psi(\{c_2, c_3\}) = c_2 \text{ and } c_3$$

$$\psi(\{c_1, c_4\}) = c_1 \text{ and } c_4$$

c.  **$\psi$  applied to  $\langle K/C_3, \leq_{C_3} \rangle$ :**

$$\psi(\{c_2\}) = c_2$$

$$\psi(\{c_1, c_4\}) = c_1 \text{ and } c_4$$

$$\psi(\{c_3, c_5\}) = c_3 \text{ and } c_5$$

We can show that the mapping  $\psi$  is an order-embedding mapping. This is stated as a theorem in (24).

---

<sup>4</sup> Note that  $\psi$  is not necessarily a function.  $\psi$  maps an equivalence class onto each of its members, and since an equivalence class can have more than one member,  $\psi$  can be a multi-valued mapping.

(24) **Theorem 3: That  $\psi$  is an order-embedding**

The mapping  $\psi$  as defined in Def. 9 (22) is an order-embedding.

Proof of Theorem 3: We need to show that  $fk_{1,C} \leq_C fk_{2,C}$  if and only if  $\psi(fk_{1,C}) \leq_{C'} \psi(fk_{2,C})$ . First consider the *if* part – i.e. I will first show that  $\psi(fk_{1,C}) \leq_{C'} \psi(fk_{2,C})$  implies  $fk_{1,C} \leq_C fk_{2,C}$ . Both  $\psi(fk_{1,C})$  and  $\psi(fk_{2,C})$  are members of the set  $\langle K, \leq_{C'} \rangle$  – since  $\psi$  is defined as mapping into this set (Def. 9 (22)). If  $\psi(fk_{1,C}) \leq_{C'} \psi(fk_{2,C})$ , then by the definition of  $\leq_{C'}$  (Def. 7 (19)) it follows that  $C(\psi(fk_{1,C})) \leq C(\psi(fk_{2,C}))$ . But then by definition of the ordering  $\leq_C$  (Def. 6 (17)), it follows directly that  $fk_{1,C} \leq_C fk_{2,C}$ .

Now consider the *only* part – i.e. I will now show that  $fk_{1,C} \leq_C fk_{2,C}$  only if  $\psi(fk_{1,C}) \leq_{C'} \psi(fk_{2,C})$ . To show this, assume the opposite – i.e. assume that  $fk_{1,C} \leq_C fk_{2,C}$  but that  $\psi(fk_{1,C}) >_{C'} \psi(fk_{2,C})$ .<sup>5</sup> Again, we know that both  $\psi(fk_{1,C})$  and  $\psi(fk_{2,C})$  are members of the set  $\langle K, \leq_{C'} \rangle$  – since  $\psi$  is defined as mapping into this set (Def. 9 (22)). Based on the definition of  $\leq_{C'}$  (Def. 7 (19)) we therefore know that if  $\psi(fk_{1,C}) >_{C'} \psi(fk_{2,C})$ , then  $C(\psi(fk_{1,C})) > C(\psi(fk_{2,C}))$ . But based on the definition of  $\leq_C$  (Def. 6 (17)),  $C(\psi(fk_{1,C})) > C(\psi(fk_{2,C}))$  implies  $fk_{1,C} >_C fk_{2,C}$ . And this contradicts the assumption that we started with, i.e. that  $fk_{1,C} \leq_C fk_{2,C}$ . So, this means that  $fk_{1,C} \leq_C fk_{2,C}$  only if  $\psi(fk_{1,C}) \leq_{C'} \psi(fk_{2,C})$ .

□

This property of  $\psi$  is clear in the examples in (23). Consider  $C_1$  as an example.

We know from (18) that  $\{c_3\} \leq_{C_1} \{c_1, c_4\}$ , and from (20) we know that  $c_3 \leq_{C_1} c_1$ ,  $c_3 \leq_{C_1} c_4$ ,

---

<sup>5</sup> If  $\neg(\psi(fk_{1,C}) \leq_{C'} \psi(fk_{2,C}))$  there are two possibilities to consider. It is possible that  $\psi(fk_{1,C}) >_{C'} \psi(fk_{2,C})$  but it is also possible that  $\psi(fk_{1,C})$  and  $\psi(fk_{2,C})$  not comparable in terms of the relation  $\leq_{C'}$ . Because of the fact that constraints are functions, all candidates are comparable in terms of all constraints so that  $\neg(\psi(fk_{1,C}) \leq_{C'} \psi(fk_{2,C}))$  necessarily implies  $\psi(fk_{1,C}) >_{C'} \psi(fk_{2,C})$ .

and  $c_1 \leq_{C^1} c_4$ . In (23) we see that  $\psi(\{c_3\}) = c_3$  and  $\psi(\{c_1, c_4\}) = c_1$  and  $c_4$ . Therefore, we have  $\{c_3\} \leq_{C^1} \{c_1, c_4\}$  and  $\psi(\{c_3\}) \leq_{C^1} \psi(\{c_1, c_4\})$ . It can easily be checked that this true for all candidates and all three constraints.

What we have shown is that there exists an order-embedding mapping from  $\langle K/C, \leq_C \rangle$  to  $\langle K, \leq_{C'} \rangle$ . It is therefore a straightforward matter to move from the ordered set of equivalence classes  $\langle K/C, \leq_C \rangle$  to the ordered set of candidates  $\langle K, \leq_{C'} \rangle$ . Consequently, it does not matter in principle whether we think of the ordering that EVAL imposes on the candidate set in terms of individual candidates ( $\leq_{C'}$ ) or in terms of equivalence classes of candidates ( $\leq_C$ ) – the one can always be recovered from the other. In the rest of this chapter I will deal with the ordering only in terms of equivalence classes ( $\leq_C$ ). The reason for this is that the ordering in terms of equivalence classes ( $\leq_C$ ) is considerably simpler than the ordering in terms of individual candidates ( $\leq_{C'}$ ). For one thing, the set  $K/C$  contains potentially fewer candidates than the set  $K$  – since every member of  $K/C$  can contain several members of  $K$ . There are therefore fewer ordering relations to consider in the set  $\langle K/C, \leq_C \rangle$  than in the set  $\langle K, \leq_{C'} \rangle$ .<sup>6</sup>

This issue of the relationship between equivalence classes and individual candidates will also crop in a different guise towards the end of this chapter where I discuss the concept of “grammatical distinctness”. Two candidates that belong to the

---

<sup>6</sup> There are more reasons. Strictly speaking  $\leq_{C'}$  is not even an order, but only a “pre-order” or a “quasi-order” – i.e. it is a transitive and reflexive but not antisymmetric relation (Partee *et al.*, 1993:207-208). Because of this many of the properties that we know to hold of orders in general do not necessarily hold of  $\leq_{C'}$ . Working with  $\leq_C$ , which is an order, is therefore more convenient – it allows us to use all of the standard concepts and theorems that apply to orders.

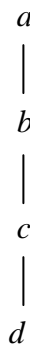
same equivalence class in terms of constraint  $C$  are “grammatically indistinct” in terms  $C$  (see §3.2.5 for more on this).

### 2.2.2 $\leq_C$ defines a chain

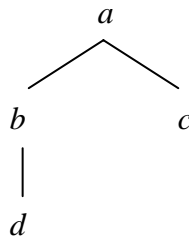
In this section I will show that the ordering that  $\leq_C$  imposes on  $K/C$  is a chain ordering. Not all orderings are chain orderings. For an order to qualify as a chain ordering, it is necessary that any two elements be comparable. It is easiest to show the difference graphically. In (25) I show a graphic representation of three orderings on the set  $A = \{a, b, c, d\}$  of which only the first is a chain.

#### (25) Chains and non-chains

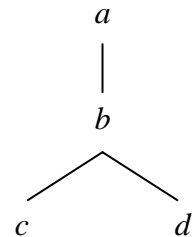
**A chain ordering**



**Non-chain I**



**Non-chain II**



The second ordering in (25) is not a chain, because  $b$  and  $c$  are not comparable in this ordering, and neither is  $c$  and  $d$ . Similarly, the third ordering is not a chain since not all elements are comparable – here  $c$  and  $d$  are not comparable.

In (26) I formally define a chain (Davey and Priestley, 1990:3), and in (27) then state explicitly that  $\leq_C$  defines a chain on  $K/C$ .

#### (26) **Def. 10: Definition of a chain**

Let  $P$  be an ordered set. Then  $P$  is a chain iff for all  $x, y \in P$ , either  $x \leq y$  or  $y \leq x$ .



(27) **Theorem 4: That  $\leq_C$  defines a chain**

The ordering that  $\leq_C$  imposes on  $K/C$  is a chain.

Proof of Theorem 4: Consider  $fk_{1,C}, fk_{2,C} \in K/C$ , with  $fk_{1,C}$  and  $fk_{2,C}$  not necessarily distinct. By trichotomy of  $\leq$  on  $\cup$  (Enderton, 1977:62-63), it follows that one of the following is true:  $C(k_1) < C(k_2)$ ,  $C(k_1) > C(k_2)$ , or  $C(k_1) = C(k_2)$ . By the definition of  $\leq_C$  (Def. 6 (17)), it then follows directly that either  $fk_{1,C} >_C fk_{2,C}$ ,  $fk_{1,C} <_C fk_{2,C}$  or  $fk_{1,C} =_C fk_{2,C}$ . □

Looking back at the orderings in (18) that is associated with each of the three constraints in our example, it is clear that these three orderings are indeed chains. In each quotient set  $K/C_i$  all equivalence classes are comparable by the relation  $\leq_{C_i}$ .

What implications does this have for an OT grammar? It follows from this that there is no indeterminacy in the order that  $\leq_C$  imposes on  $K/C$ . It is always possible to determine for any two equivalence classes in  $K/C$  how they are related with regard to each other in terms of  $\leq_C$ . And because of the natural relationship between  $\langle K/C, \leq_C \rangle$  and  $\langle K, \leq_C \rangle$  (see §2.2.1 just above), it is also possible to determine for any two candidates how they are related to each other in terms of  $\leq_C$ . For any constraint  $C$  and any two candidates  $k_1$  and  $k_2$ , we can therefore determine from the set  $\langle K/C, \leq_C \rangle$  whether  $C(k_1) < C(k_2)$ ,  $C(k_1) > C(k_2)$  or  $C(k_1) = C(k_2)$ .<sup>7</sup> This point will be discussed again when I consider the ordering imposed on the candidate set with respect to the full grammar (§3.2.2).

---

<sup>7</sup> In this sense the classic OT model, and specifically the model developed here, is different from OT with targeted constraints (Bakovic and Wilson, 2000, Wilson, 1999). A targeted constraint takes only a subset of the candidate set as domain. For a targeted constraint, EVAL can therefore impose an ordering only on those candidates in the domain of the constraint. Candidates not in the domain of a targeted constraint cannot be related to other candidates in terms of the ordering that EVAL imposes

### 2.2.3 The chain defined by $\leq_C$ always has a minimum

In this section I will show that the chain ordering that  $\leq_C$  defines on the set  $K/C$  is guaranteed to have a minimum. The minimum will be that equivalence class that contains the candidates that receive the smallest number of violations in terms of  $C$ . The definition in (28) is from Davey and Priestley (1990:15).

(28) **Def. 11: Minimum of an ordered set**

Let  $P$  be an ordered set and  $Q \subseteq P$ . Then:

$a \in Q$  is the minimum of  $Q$  iff  $a \leq x$  for every  $x \in Q$ .

(29) **Theorem 5: That  $\langle K/C, \leq_C \rangle$  has a minimum**

The ordering  $\leq_C$  always has a minimum in  $K/C$ .

Proof of Theorem 5:<sup>8</sup> This proof makes use of the notion of the *well-ordering* of  $\mathbb{N}$  under  $\leq$  (Enderton, 1977:86). We say that  $\mathbb{N}$  is well-ordered under  $\leq$ , because every non-empty subset of  $\mathbb{N}$  is guaranteed to have a minimum under  $\leq$ .<sup>9</sup>

Constraints are functions with their ranges included in  $\mathbb{N}$  – that is for all candidates  $k$  and all constraints  $C$ ,  $C(k) \in \mathbb{N}$ . By the well-ordering of  $\mathbb{N}$  under  $\leq$ , it follows that there will be some candidate  $k$  such that  $C$  will map  $k$  onto a smaller number than all other candidates, i.e. there will be some  $k$  such that  $C(k) \leq C(k')$  for all  $k' \in K$ . In

---

on the candidate set. The orderings associated with targeted constraints are therefore not chain orderings.

<sup>8</sup> A minimum is defined on a subset  $Q$  of the ordered set  $P$ . In the case under consideration here, the subset is equal to the superset – that is,  $K/C$  stands for both  $P$  and  $Q$  from the definition (which is possible since  $K/C \subseteq K/C$ ). The proof therefore does not refer to the subset~superset relation.

<sup>9</sup> This is obviously true – in any set of natural numbers there will always be a smallest number. It can also be proved formally. See Enderton (1977:86-87) for a proof.

terms of the ordering  $\leq_C$  (Def. 6 (17)) the equivalence class of  $k$ ,  $fk_{,C}$  (Def. 4 (13)), will then precede all other equivalence classes in the quotient set  $K/C$  (Def. 5 (15)), i.e.  $fk_{,C} \leq_C fk'_{,C}$  for all  $fk'_{,C} \in K/C$ . And therefore  $fk_{,C}$  is then the minimum in  $K/C$ .  $\square$

The orderings in (18) that are associated with each of the three constraints in our example clearly all have a minimum – in each of the orderings it is the equivalence class that appears highest on the graphic representations of the orderings. In this example this is every time the equivalence class containing the candidates that receive zero violations in terms of the specific constraint.

The converse of Theorem 5 is of course not necessarily true. Since there are constraints that can in principle assign an unbounded number of violations,<sup>10</sup> it is possible that the chain imposed by  $\leq_C$  on  $K/C$  can be without a maximum. If this chain had no maximum and no minimum, then there would have been no uniquely identifiable point on the chain – for any member of the chain there would be infinitely many members above it and also infinitely members below it. The output of an OT grammar would not have been very informative had it been such an infinitely ascending and infinitely descending chain. We would never be able to refer uniquely to a specific level in the chain, and it would therefore not be possible to define access to the chain, i.e. access to the candidate set. I will return to this point again in §3.2.3 where I deal with the ordering imposed on the candidate set with reference to the full constraint set.

---

<sup>10</sup> There is nothing that limits the number of epenthetic segments in principle, so that DEP can assign an unbounded number of violations. There are also many markedness constraints that are not in principled limited in how many violations they can assign. Those markedness constraints that penalize a candidate for each instantiation of some marked structure (ONSET, NOCODA, \*[+voice, +obstruent], etc.) can assign unboundedly many violations in principle – since unboundedly many violating structures can be inserted.

### 2.3.4 The set $K/C$ is a partition on $K$

In this section I will show that the set  $K/C$  is a partition on  $K$ . What this means is that every member of  $K$  (every candidate) is included in one and exactly one of the equivalence classes contained in  $K/C$ . This is an important result for two reasons: First, since every candidate is included in some equivalence class, it means that the ordering  $\leq_C$  does indeed give us information about every candidate. Secondly, every candidate is included in only one equivalence class, and every equivalence class can occupy only one position in the ordering that  $\leq_C$  imposes on  $K/C$ . Because of the natural relationship between  $\langle K/C, \leq_C \rangle$  and  $\langle K, \leq_C \rangle$  (see §2.2.1), this implies that every candidate is also guaranteed to occupy a unique slot in the ranking that EVAL imposes on the candidate set. The fact that  $K/C$  is a partition on  $K$  therefore assures that we can determine for every candidate what its unique relationship is to every other candidate in terms of the constraint  $C$ .

The definition of a partition in (30) is based on Enderton (1977:57) and Partee *et al.* (1993:46).

(30) **Def. 12: A partition**

A set  $P$  is said to be a partition on some set  $A$  iff:

- (a)  $P$  consists of non-empty subsets of  $A$ .
- (b) The sets in  $P$  are exhaustive – each element of  $A$  is in some set in  $P$ .
- (c) The sets in  $P$  are disjoint – no two different sets in  $P$  have any element in common.

(31) **Theorem 6:  $K/C$  as a partition on  $K$**

$K/C$  is a partition on  $K$ .

Proof of Theorem 6: Consider first (a), the requirement that the sets in  $K/C$  be non-empty.  $K/C$  is the quotient set on  $K$  modulo  $\approx_C$  (Def. 5 (15)). Every member of  $K/C$  is therefore an equivalence class on  $K$  under the equivalence relation  $\approx_C$  (Def. 4 (13)). Equivalence relations are reflexive (Def. 3 (11)), and therefore an equivalence class can never be empty. For any member  $fk_{,C}$  of  $K/C$  it then follows that  $fk_{,C}$  has at least one member, namely  $k$  (since  $k \approx_C k$ ).

Now consider requirement (b), that the equivalence classes be exhaustive. Since constraints are functions with  $K$  as their domain ((3) and Theorem 1 (5)), it follows that  $C(k)$  is defined for every  $k \in K$ . Then we have for every  $k \in K$ ,  $C(k) = C(k)$ , and therefore  $k \approx_C k$  (Def. 2 (9)). And finally we have for every  $k \in K$ ,  $k \in fk_{,C} \in K/C$  (Def. 4 (13) and Def. 5 (15)).

Now consider requirement (c), that the equivalence classes be disjoint. Let  $k_1, k_2, k_3 \in K$  and let  $fk_{2,C}, fk_{3,C}$  be equivalence classes associated with  $k_2$  and  $k_3$  respectively, as defined in Def. 4 (13). Now let  $k_1 \in fk_{2,C}$  and  $k_1 \in fk_{3,C}$ . We need to show that  $fk_{2,C} = fk_{3,C}$ .

Since  $k_1 \in fk_{2,C}$  and  $k_1 \in fk_{3,C}$ , we know that  $k_2 \approx_C k_1$  and  $k_3 \approx_C k_1$  (Def. 4 (13)). But  $\approx_C$  is an equivalence relation (Theorem 2 (12)), and therefore  $\approx_C$  is symmetric (Def. 3 (11)). Then we have  $k_1 \approx_C k_3$ , because of  $k_3 \approx_C k_1$ . But as an equivalence relation  $\approx_C$  is also transitive (Def. 3 (11)). And therefore  $k_2 \approx_C k_1$  and  $k_1 \approx_C k_3$  implies  $k_2 \approx_C k_3$ . But again

since  $\approx_C$  is transitive, it follows from  $k_2 \approx_C k_3$  that for all  $k$  such that  $k_3 \approx_C k$ , also  $k_2 \approx_C k$ . And therefore for all  $k$ , if  $k \in fk_{3,C}$ , then  $k \in fk_{2,C}$  (Def. 4 (13)). By similar reasoning we can show the converse, that for all  $k$ , if  $k \in fk_{2,C}$ , then  $k \in fk_{3,C}$ . Therefore we have  $fk_{2,C} = fk_{3,C}$ . □

In (16) the quotient sets associated with each of the constraints used in our examples were listed. Inspection of these equivalence classes will show that each of them are indeed partitions on  $K$  – every candidate is included in one and only one equivalence class in every quotient set.

We therefore now know that every candidate is included in exactly one of the equivalence classes that make up  $K/C$ . Together with the fact that  $\leq_C$  defines a chain ordering on  $K/C$  and with the fact that there is a natural relationship between  $\langle K/C, \leq_C \rangle$  and  $\langle K, \leq_C \rangle$  (see §2.2.1), this implies that every candidate has one unique spot in the ordering that EVAL imposes on the candidate set. From the information in the ordered set  $\langle K/C, \leq_C \rangle$  we can determine for any two distinct candidates  $k_1$  and  $k_2$  how they are related to each other in terms of constraint  $C$  – i.e. whether they fare equally well on  $C$  ( $C(k_1) = C(k_2)$ ), or whether one does better on  $C$  ( $C(k_1) < C(k_2)$  or  $C(k_1) > C(k_2)$ ). There is no indeterminacy in the output of an OT grammar. I will return to this point again in §3.2.4 where the ordering imposed on the candidate set with reference to the full constraint set is discussed.

### 3. EVAL and the ordering associated with the full constraint set

Up to this point I have considered only how EVAL orders the candidate set with respect to individual constraints. Once EVAL has done this for every constraint in CON, we end up with as many orderings on the candidate set as there are constraints in CON. But the final output of the grammar is a single ordering on the candidate set. We therefore need a way in which these different orderings can be combined into one single ordering that corresponds to the whole grammar. This section of the chapter deals with how EVAL combines the orderings associated with different constraints.

To make the problem that needs to be addressed more concrete I will continue using the same example as earlier. I repeat the tableau from (6) in (32).

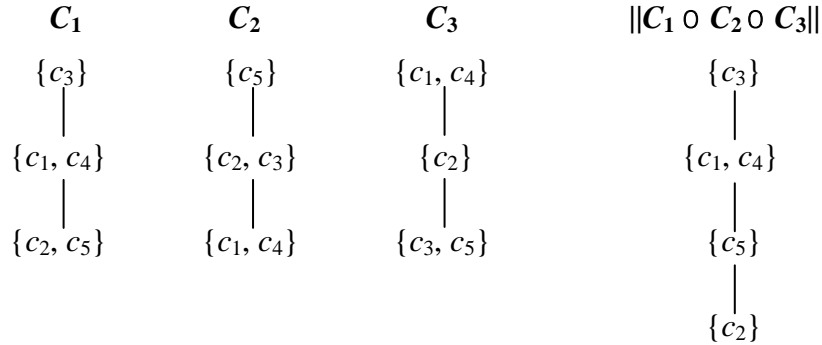
(32)  $\{c_1, c_2, c_3, c_4, c_5\}$  evaluated by  $\|C_1 \circ C_2 \circ C_3\|$

	$C_1$	$C_2$	$C_3$
$c_1$	*	**	
$c_2$	**	*	*
$c_3$		*	**
$c_4$	*	**	
$c_5$	**		**

In this grammar,  $c_3$  is clearly the best candidate. If we remove  $c_3$  from the candidate set, and consider only the four remaining candidates, then  $c_1$  and  $c_4$  tie as the best. If we also remove these candidates and consider only the two remaining candidates  $\{c_2, c_5\}$ , then  $c_5$  is the best. In this grammar the candidates are therefore related as follows in terms of their harmony:  $|c_3 \text{ } ^{\text{TM}} \{c_1, c_4\} \text{ } ^{\text{TM}} c_5 \text{ } ^{\text{TM}} c_2|$ . The ordering that EVAL imposes on the candidate set in terms of each individual constraint, and in terms of the grammar as a

whole can then be represented graphically as in (33). The orderings relative to the individual constraints are, of course, identical to the orderings shown above in (18).

(33) **Orderings on the candidate set relative to  $C_1, C_2, C_3$ , and  $\|C_1 \circ C_2 \circ C_3\|$**



This makes it clear what we need: a way in which to combine the orderings associated with each of  $C_1, C_2$  and  $C_3$  such that the ordering that results is the ordering associated with  $\|C_1 \circ C_2 \circ C_3\|$ . Section §3.1 is dedicated to showing how this goal can be achieved. This combination process is defined in two stages. First, the Cartesian product is taken between the quotient sets associated with individual constraints (§3.1.1), and the set that results from this process is ordered lexicographically (§3.1.2). Then this new set and ordering is simplified by applying set intersection to each member of this new set (§3.1.3). The simplified set and ordering is the final output of the grammar, the final ordering that EVAL imposes on the candidate set for the grammar as a whole. Section §3.2 considers some of the properties of this set and ordering.

### 3.1 Combining the ordered sets associated with individual constraints

#### 3.1.1 Taking the Cartesian product

We need a way in which to combine the ordered sets associated with each of the constraints. There are several different ways in which ordered sets can be joined – see Davey and Priestly (1990:17-19) for a discussion of the most important ways in which



this can be done. One way in which ordered sets can be combined is by taking the Cartesian product between them. When we take the Cartesian product between two sets a new set results that contains ordered pairs  $\langle x_1, x_2 \rangle$  with  $x_1$  coming from the first set and  $x_2$  from the second set. The desirable property of this procedure is that a precedence relationship is established between the information coming from the two sets – the elements from the first set precede the elements from the second set in the ordered pairs  $\langle x_1, x_2 \rangle$ . In an OT grammar constraints are ranked and higher ranked constraints take precedence over lower ranked constraints. We have to take this fact in consideration when we combine the orderings associated with individual constraints into one conglomerate ordering for the grammar as a whole. The Cartesian product operation allows us to do exactly this – we take the Cartesian product of the ordered sets associated with each constraint in the order in which the constraints are ranked. This section explains how this can be achieved formally.

We usually think of the Cartesian product as the product between two sets. However, there is nothing that prohibits us from taking the Cartesian product of more than two sets. The Cartesian product of two sets is a set of ordered pairs  $\langle x_1, x_2 \rangle$  with  $x_1$  coming from the first set and  $x_2$  from the second set. When we take the Cartesian between  $n$  sets, the result is a set of  $n$ -tuples  $\langle x_1, x_2, \dots, x_n \rangle$  with  $x_i$  coming from the  $i$ th set. The definition of the Cartesian product in (34) is based on Enderton (1977:54). Simply restating this definition in terms of quotient sets associated with constraints, gives us the first step in the combination process. This is done in (35).

(34) **Def. 13: Cartesian product**

Let  $I$  be the set  $\{1, 2, \dots, n\}$ , the *index set*, and let  $H$  be a function with domain  $I$ .

Then, for each  $i \in I$ , we have a set  $H(i)$ . The Cartesian product of  $H(i)$  for all  $i \in I$  is defined as follows:

$$\mathbf{V}_{i \in I} H(i) := \{f \mid f \text{ is a function with domain } I \text{ and } \forall i (i \in I \rightarrow f(i) \in H(i))\}^{11}$$

(35) **Def. 14: Step 1 in combination process = Cartesian product between sets  $K/C_i$**

Let  $I$  be the set  $\{1, 2, \dots, n\}$ , the *index set*, such that  $\|C_1 \circ C_2 \circ \dots \circ C_n\|$ .

Let  $K/C_i$  be the quotient on  $K$  associated with  $C_i$ . We want the Cartesian product of all the quotient sets. We define this as follows:

$$\mathbf{V}_{i \in I} K/C_i := \{f \mid f \text{ is a function with domain } I \text{ and } \forall i (i \in I \rightarrow f(i) \in K/C_i)\}$$

The set  $\mathbf{V}_{i \in I} K/C_i$  will be referred to as  $K/C_\times$ .

To see what the result of this process is, consider again the example from (6) and (32) above. In (16) the quotient sets associated with each of the three constraints were listed. In (36) I show the result of taking the Cartesian product between these three sets in the order  $K/C_1 \times K/C_2 \times K/C_3$ .

---

<sup>11</sup> Each  $f \in \mathbf{V}_{i \in I} H(i)$  is of the following form:  $f = \{\langle 1, x_1 \rangle, \langle 2, x_2 \rangle \dots \langle n, x_n \rangle\}$  with  $x_i \in H(i)$ . The set that we actually want is a set whose members are ordered  $n$ -tuples, i.e. of the form  $\langle x_1, x_2 \dots x_n \rangle$  with  $x_i \in H(i)$ . However, it is a straightforward matter to uniquely match up every set  $\{\langle 1, x_1 \rangle, \langle 2, x_2 \rangle \dots \langle n, x_n \rangle\}$  with that  $n$ -tuple  $\langle x_1, x_2 \dots x_n \rangle$  to which it corresponds. The  $n$ -tuple that corresponds to the set  $\{\langle 1, x_1 \rangle, \langle 2, x_2 \rangle \dots \langle n, x_n \rangle\}$  is namely that  $n$ -tuple in which  $x_i < x_j$  iff  $\langle i, x_i \rangle, \langle j, x_j \rangle \in \{\langle 1, x_1 \rangle, \langle 2, x_2 \rangle \dots \langle n, x_n \rangle\}$ , and  $i < j$ . Since this uniquely matches up the sets  $\{\langle 1, x_1 \rangle, \langle 2, x_2 \rangle \dots \langle n, x_n \rangle\}$  with their corresponding  $n$ -tuples  $\langle x_1, x_2 \dots x_n \rangle$ , we can without loss of exactness use the set notation and the  $n$ -tuple notation interchangeably.

(36) **Taking the Cartesian product of  $K/C_1$ ,  $K/C_2$  and  $K/C_3$  from (17)**

$$K/C_1 = \{\{c_3\}, \{c_1, c_4\}, \{c_2, c_5\}\}$$

$$K/C_2 = \{\{c_5\}, \{c_2, c_3\}, \{c_1, c_4\}\}$$

$$K/C_3 = \{\{c_1, c_4\}, \{c_2\}, \{c_3, c_5\}\}$$

$$K/C_x =$$

$$\{ \langle \{c_3\}, \{c_5\}, \{c_1, c_4\} \rangle, \quad \langle \{c_3\}, \{c_5\}, \{c_2\} \rangle, \quad \langle \{c_3\}, \{c_5\}, \{c_3, c_5\} \rangle, \\ \langle \{c_3\}, \{c_2, c_3\}, \{c_1, c_4\} \rangle, \quad \langle \{c_3\}, \{c_2, c_3\}, \{c_2\} \rangle, \quad \langle \{c_3\}, \{c_2, c_3\}, \{c_3, c_5\} \rangle, \\ \langle \{c_3\}, \{c_1, c_4\}, \{c_1, c_4\} \rangle, \quad \langle \{c_3\}, \{c_1, c_4\}, \{c_2\} \rangle, \quad \langle \{c_3\}, \{c_1, c_4\}, \{c_3, c_5\} \rangle, \\ \langle \{c_1, c_4\}, \{c_5\}, \{c_1, c_4\} \rangle, \quad \langle \{c_1, c_4\}, \{c_5\}, \{c_2\} \rangle, \quad \langle \{c_1, c_4\}, \{c_5\}, \{c_3, c_5\} \rangle, \\ \langle \{c_1, c_4\}, \{c_2, c_3\}, \{c_1, c_4\} \rangle, \quad \langle \{c_1, c_4\}, \{c_2, c_3\}, \{c_2\} \rangle, \quad \langle \{c_1, c_4\}, \{c_2, c_3\}, \{c_3, c_5\} \rangle, \\ \langle \{c_1, c_4\}, \{c_1, c_4\}, \{c_1, c_4\} \rangle, \quad \langle \{c_1, c_4\}, \{c_1, c_4\}, \{c_2\} \rangle, \quad \langle \{c_1, c_4\}, \{c_1, c_4\}, \{c_3, c_5\} \rangle, \\ \langle \{c_2, c_5\}, \{c_5\}, \{c_1, c_4\} \rangle, \quad \langle \{c_2, c_5\}, \{c_5\}, \{c_2\} \rangle, \quad \langle \{c_2, c_5\}, \{c_5\}, \{c_3, c_5\} \rangle, \\ \langle \{c_2, c_5\}, \{c_2, c_3\}, \{c_1, c_4\} \rangle, \quad \langle \{c_2, c_5\}, \{c_2, c_3\}, \{c_2\} \rangle, \quad \langle \{c_2, c_5\}, \{c_2, c_3\}, \{c_3, c_5\} \rangle, \\ \langle \{c_2, c_5\}, \{c_1, c_4\}, \{c_1, c_4\} \rangle, \quad \langle \{c_2, c_5\}, \{c_1, c_4\}, \{c_2\} \rangle, \quad \langle \{c_2, c_5\}, \{c_1, c_4\}, \{c_3, c_5\} \rangle \}$$

### 3.1.2 Imposing the lexicographic order on $K/C_x$

At this point we have a set  $K/C_x$ , but this is still an unordered set. In this section I will show how we can define an ordering on this set. The lexicographic order is an ordering relationship on the product of  $n$  ordered sets that gives primacy to the order of the  $(i-1)$ th set over that of the  $i$ th set. This is desirable in OT – since the order imposed by the higher ranked constraints should be more important in the combined ordering.<sup>12</sup> The definition of the lexicographic ordering in (37) is based on Davey and Priestley (1990:19). However, they define the order only on a binary Cartesian product, while the definition in (37) is extended to cover arbitrary Cartesian products. In (38) I restate this definition in

---

<sup>12</sup> This follows from the strictness of strictness domination principle of OT (McCarthy, 2002b:4, Prince and Smolensky, 1993:78, 1997:1604). For more on this, see §3.2.1 below.

terms of the Cartesian product between the quotient sets associated with constraints – see (35) above.

(37) **Def. 15: Lexicographic order**

Let  $\mathbf{V}_{i \in I} H(i)$  be the set as defined in Def. 13 (34) above, and let  $\langle x_1, x_2, \dots, x_n \rangle, \langle y_1, y_2, \dots, y_n \rangle \in \mathbf{V}_{i \in I} H(i)$ .

The lexicographic order on  $\mathbf{V}_{i \in I} H(i)$  is defined as follows:

$\langle x_1, x_2, \dots, x_n \rangle \leq \langle y_1, y_2, \dots, y_n \rangle$  iff:

(i) For all  $i \leq n, x_i = y_i$  (then  $\langle x_1, x_2, \dots, x_n \rangle = \langle y_1, y_2, \dots, y_n \rangle$ )

OR (ii)  $\exists k$  such that:

•  $\forall i (i < k \rightarrow x_i = y_i)$ , and

•  $x_k < y_k$  (then  $\langle x_1, x_2, \dots, x_n \rangle < \langle y_1, y_2, \dots, y_n \rangle$ )

(38) **Def. 16: Step 2 in the combination process = ordering  $K/C_x$**

Let  $C_i \in \text{CON}$ , with the ranking  $\|C_1 \circ C_2 \circ \dots \circ C_n\|$ , and  $K/C_i$  the quotient set associated with constraint  $C_i$  (as defined in Def. 5 (15)). Let  $f_{x_i, C_i}, f_{y_i, C_i} \in K/C_i$  be the equivalence classes of candidates  $x_i$  and  $y_i$  in terms of constraint  $C_i$  (as defined in Def. 4 (13)).

Let  $\leq_{C_i}$  be the ordering that EVAL imposes on the candidate set in terms of constraint  $C_i$  (as defined in Def. 6 (17)).

Let  $K/C_x$  be the Cartesian product of  $K/C_i$  for all  $i \in I$  (as defined in Def. 14 (35)).

Let  $\langle f_{x_1, C_1}, f_{x_2, C_2} \dots f_{x_n, C_n} \rangle, \langle f_{y_1, C_1}, f_{y_2, C_2} \dots f_{y_n, C_n} \rangle \in K/C_x$ .

Then  $\leq_x$ , the lexicographic order on  $K/C_x$ , is defined as follows:

((38) continued)

$\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle \leq_x \langle fy_{1,C1}, fy_{2,C2} \dots fy_{n,Cn} \rangle$  iff:

(i)  $\forall i (i \leq n \rightarrow fx_{i,Ci} =_{Ci} fy_{i,Ci})$

(then  $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle =_x \langle fy_{1,C1}, fy_{2,C2} \dots fy_{n,Cn} \rangle$ )

OR (ii)  $\exists k$  such that:

•  $\forall i (i < k \rightarrow fx_{i,Ci} =_{Ci} fy_{i,Ci})$ , and

•  $fx_{k,Ck} <_{Ck} fy_{k,Ck}$ .

(then  $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle <_x \langle fy_{1,C1}, fy_{2,C2} \dots fy_{n,Cn} \rangle$ )

In order to make the discussion more concrete, I show in (39) the result of imposing the lexicographic order as defined here in (38) on the set  $K/C_x$  from (36).

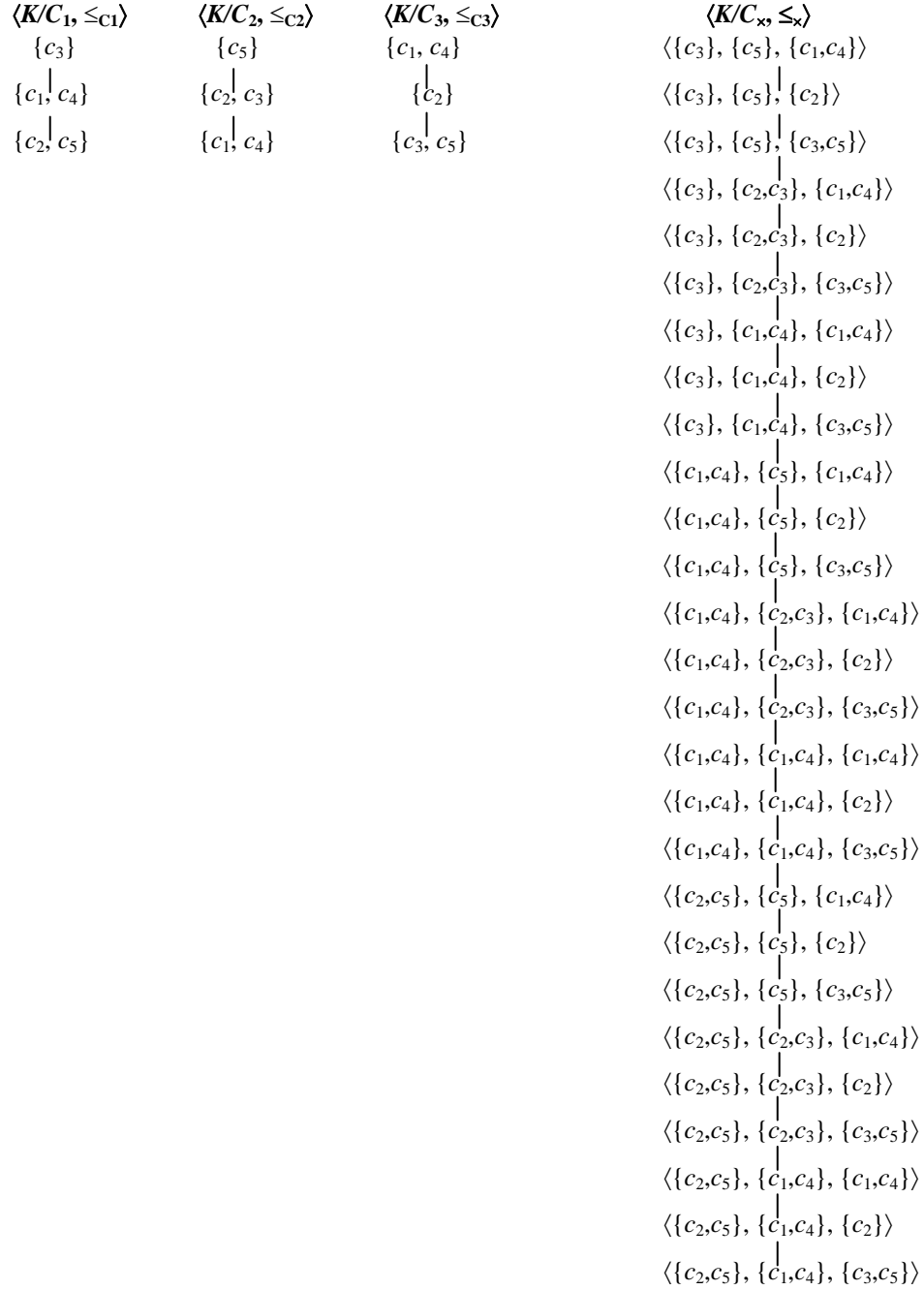
As is clear from (36) and (39), the set  $K/C_x$  is a set of  $n$ -tuples of sets, while  $K/C_1$ ,  $K/C_2$  and  $K/C_3$  are sets of sets. An OT grammar ultimately makes claims about (equivalence classes of) candidates,<sup>13</sup> and not about  $n$ -tuples of (equivalence classes of) candidates. It is therefore necessary to simplify the set  $K/C_x$  such that it is also a set of sets. The next sub-section introduces a method for doing this.

However, before we discuss this simplification process, I will first prove two lemmas about the ordering  $\leq_x$  on the set  $K/C_x$ . The ordering that EVAL imposes on the simplified set (discussed in the next section) is defined in terms of the ordering  $\leq_x$ . Although the characteristics of the ordering  $\leq_x$  are not themselves of direct relevance, these characteristics will become instrumental in the later discussion (§3.2.3 and §3.2.4).

---

<sup>13</sup> See §2.1.1 for the relationship between individual candidates and the equivalence classes of candidates.

(39) **Imposing the lexicographic order on  $K/C_x$**



(40) **Lemma 1: That  $\leq_x$  defines a chain**

$\leq_x$  defines a chain on  $K/C_x$ .<sup>14</sup>

---

<sup>14</sup> For a definition of a chain refer to Def. 10 (26) above.

Proof of Lemma 1: Let  $\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle, \langle fy_{1,C_1}, fy_{2,C_2} \dots fy_{n,C_n} \rangle \in K/C_x$ ,  $K/C_i$  the quotient set associated with constraint  $C_i$  (Def. 5 (15)), and  $\leq_{C_i}$  the ordering on this quotient set (Def. 6 (17)). Then we have  $fx_{i,C_i}, fy_{i,C_i} \in K/C_i$  for all  $i \leq n$  (Def. 14 (35)). But since  $\leq_{C_i}$  defines a chain ordering on  $K/C_i$  (Theorem 4 (27)), we have that  $fx_{i,C_i} \equiv_{C_i} fy_{i,C_i}$  or  $fx_{i,C_i} >_{C_i} fy_{i,C_i}$  or  $fx_{i,C_i} <_{C_i} fy_{i,C_i}$  for all  $i \leq n$ . The ordering  $\leq_x$  is defined in terms of the orderings  $\leq_{C_i}$ , so that it follows immediately that any two elements in  $K/C_x$  are comparable, and therefore that  $\leq_x$  orders  $K/C_x$  in a chain.  $\square$

(41) **Lemma 2: That  $\leq_x$  always has a minimum**

The ordering  $\leq_x$  always has a minimum in  $K/C_x$ .<sup>15</sup>

Proof of Lemma 2: There are two possible scenarios: (i) Either all candidates receive exactly the same number of violations in terms of all constraints, or (ii) there are at least two candidates that differ on at least one constraint. I will consider these two scenarios in turn below.

Scenario 1: All candidates receive exactly the same number of violations in terms of every constraint. If all candidates receive the same number of violations in terms some constraint  $C$ , then we have  $k_i \approx_C k_j$  for all  $k_i, k_j \in K$  (Def. 2 (9)). Then we have that all candidates will belong to the same equivalence class in terms of constraint  $C$ , i.e. for all  $k_i, k_j \in K$  we have  $k_i, k_j \in fk_{i,C}$  (Def. 4 (13)). The quotient set on  $K$  associated with  $C$ ,  $K/C$  (Def. 5 (15)), will then contain the set  $fk_{i,C}$ . But since  $K/C$  is a partition on  $K$  (Theorem 6 (31)), its members exhaust all the candidates in  $k$  and are disjoint (Def. 12 (30)). Therefore,  $K/C$  contains only one member, namely  $fk_{i,C}$ .

---

<sup>15</sup> For a definition of a minimum see Def. 11 (28) above.

Since all candidates receive the same number of violations in terms of all constraints, it follows that the quotient set associated with each constraint will have only one member, i.e. if there are  $n$  constraints, then for all  $i \leq n$ ,  $K/C_i$  will contain only one member. If we take the Cartesian product of any number of sets each with only one member, the result will also be a set with only one member (Def. 13 (34)). The set  $K/C_x$  will therefore also contain only member. And then  $\leq_x$  defines a one-level chain, so that the single level on this chain is obviously the minimum of the chain.

Scenario 2: There are at least two candidates that differ on at least one constraint. I will assume that  $\leq_x$  does not have a minimum in  $K/C_x$ , and then show that it leads to a contradiction.

Assume that  $\leq_x$  has no minimum in  $K/C_x$ . Since  $\leq_x$  is a chain (Lemma 1 (40)), it then follows that  $\leq_x$  is an infinitely descending chain.

Let  $\|C_1 \circ C_2 \circ \dots \circ C_n\|$  be the grammar under consideration. Then there must be a highest ranked constraint that does not rate all candidates equally (since under assumption there are at least two candidates that differ on at least one constraint). Call this constraint  $C_i$ . Then  $\leq_{C_i}$  is the ordering associated with  $C_i$  as defined in Def. 6 (17), and  $K/C_i$  the quotient set associated with  $C_i$  as defined in Def. 5 (15). By assumption there are at least two candidates,  $k_1$  and  $k_2$ , that are rated differently by  $C_i$ , i.e.  $C_i(k_1) \neq C_i(k_2)$ . From this it follows that  $\leq_{C_i}$  establishes at least a two-level ordering on  $K/C_i$ .  $\leq_{C_i}$  defines a chain (Theorem 4 (27)) with a minimum on  $K/C_i$  (Theorem 5 (29)). So, there must be some member  $fx_{j,C_i}$  of  $K/C_i$  such that  $fx_{j,C_i} <_{C_i} fk_{r,C_i}$  for all  $fk_{r,C_i} \in K/C_i$  and  $fk_{r,C_i} \neq fx_{j,C_i}$  (i.e.  $fx_{j,C_i}$  is the minimum of  $\geq_{C_i}$  in  $K/C_i$ ).



All members of  $K/C_x$  are  $n$ -tuples of the form  $\langle f_{x,C_1}, f_{y,C_2} \dots f_{z,C_n} \rangle$  with  $x, y, z \in K$  (Def. 14 (35)). Remember that  $C_i$  is the highest ranked constraint in terms of which any two candidates differ, and  $K/C_i$  is the quotient set associated with this constraint. Each  $n$ -tuple in  $K/C_x$  will have as its  $i$ th member a member from  $K/C_i$ . Also recall that  $f_{x_j,C_i}$  is the minimum of  $\leq_{C_i}$  in  $K/C_i$ . Those  $n$ -tuples in  $K/C_x$  that have  $f_{x_j,C_i}$  as their  $i$ th member, will therefore be ordered before all  $n$ -tuples in  $K/C_x$  that have some other member of  $K/C_i$  as  $i$ th member in terms of the ordering  $\leq_x$  (Def. 16 (38)).

But by assumption  $\leq_x$  is an infinitely ascending chain. There must therefore be some  $n$ -tuple in  $K/C_x$  that is ordered before those  $n$ -tuples with  $f_{x_j,C_i}$  as their  $i$ th member. But this is not possible, since  $n$ -tuples with  $f_{x_j,C_i}$  as their  $i$ th member have just been shown to be ordered before all other  $n$ -tuples in  $K/C_x$ .

Therefore, also in the second scenario  $\leq_x$  is guaranteed to have a minimum in  $K/C_x$ . □

Now that these two lemmas have been proved, we can return to the main discussion. The next section shows how the set  $K/C_x$  and the ordering  $\leq_x$  can be simplified.

### 3.1.3 Simplifying the set $K/C_x$ and the ordering $\leq_x$

The ordering that EVAL will impose on the candidate set in our example is represented in (33). This ordering contains four members only, and each of its members is simply a set of candidates. This ordered set represents the final output of the grammar in our example. The ordered set  $\langle K/C_x, \leq_x \rangle$  represented in (39) above is very different. It has 27

members, and its members are not simply sets of candidates, but ordered  $n$ -tuples of sets of candidates. We need to find a way to transform the set  $\langle K/C_x, \leq_x \rangle$  into the set represented in (33). This section discusses the way in which we can simplify the set  $K/C_x$  and the ordering  $\leq_x$  into the set represented in (33). This transformation is defined in two steps.

I will first show how the set  $K/C_x$ , a set of ordered  $n$ -tuples of sets, can be transformed into a set of sets. There is a straightforward way to create a set of sets of candidates by simplifying  $K/C_x$  – map each  $n$ -tuple from  $K/C_x$  onto the intersection between the members of the  $n$ -tuple. One limitation that needs to be placed on this mapping, is that it must not be able to map any element from  $K/C_x$  onto  $\emptyset$ . This mapping must therefore be undefined for elements of  $K/C_x$  where the intersection of the members of the  $n$ -tuple is  $\emptyset$ .<sup>16</sup> I will first define and illustrate this mapping. Later in §3.2 I will show that this mapping has the desired properties – i.e. that applying this mapping to  $K/C_x$  in (39) will indeed result in set represented in (33).

(42) **Def. 17:** First half of step 3 in the combination process = *Intersect*

Let  $K/C_x$  be the set as defined in Def. 14 (35) above, and let  $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle \in K/C_x$ . Then we define *Intersect*:  $K/C_x \rightarrow \wp(K)$  as follows:

*Intersect*( $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle$ ) is undefined if  $fx_{1,C1} \cap fx_{2,C2} \cap \dots \cap fx_{n,Cn} = \emptyset$ ,

and

*Intersect*( $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle$ ) =  $fx_{1,C1} \cap fx_{2,C2} \cap \dots \cap fx_{n,Cn}$  otherwise.

---

<sup>16</sup> The reason for this is that we want the new set of sets that results from this simplification to be a partition on the candidate set  $K$ , and the empty set cannot be a member of any partition – see Def. 12 (30). As for why the new set should be a partition on  $K$ , see §3.2.4 below.

Based on this mapping it is possible to define a new set that contains all of those sets onto which *Intersect* does indeed map some element of  $K/C_x$ .

(43) **Def. 18: Collecting the output of *Intersect* into one set**

$K/C_{\text{Com}} := \{Z \mid \exists \langle f_{x_1, c_1}, f_{x_2, c_2} \dots f_{x_n, c_n} \rangle \in K/C_x, \text{ such that}$

$$Z = \text{Intersect}(\langle f_{x_1, c_1}, f_{x_2, c_2} \dots f_{x_n, c_n} \rangle) \}$$

To see how this operation works, consider what it does to the set  $K/C_x$  from (36) and (39) above.

(44) ***Intersect* applied to  $K/C_x$**

$\text{Intersect}(\langle \{c_3\}, \{c_5\}, \{c_1, c_4\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_3\}, \{c_5\}, \{c_2\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_3\}, \{c_5\}, \{c_3, c_5\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_3\}, \{c_2, c_3\}, \{c_1, c_4\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_3\}, \{c_2, c_3\}, \{c_2\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_3\}, \{c_2, c_3\}, \{c_3, c_5\} \rangle)$	$= \{c_3\}$
$\text{Intersect}(\langle \{c_3\}, \{c_1, c_4\}, \{c_1, c_4\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_3\}, \{c_1, c_4\}, \{c_2\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_3\}, \{c_1, c_4\}, \{c_3, c_5\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_1, c_4\}, \{c_5\}, \{c_1, c_4\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_1, c_4\}, \{c_5\}, \{c_2\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_1, c_4\}, \{c_5\}, \{c_3, c_5\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_1, c_4\}, \{c_2, c_3\}, \{c_1, c_4\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_1, c_4\}, \{c_2, c_3\}, \{c_2\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_1, c_4\}, \{c_2, c_3\}, \{c_3, c_5\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_1, c_4\}, \{c_1, c_4\}, \{c_1, c_4\} \rangle)$	$= \{c_1, c_4\}$
$\text{Intersect}(\langle \{c_1, c_4\}, \{c_1, c_4\}, \{c_2\} \rangle)$	is undefined
$\text{Intersect}(\langle \{c_1, c_4\}, \{c_1, c_4\}, \{c_3, c_5\} \rangle)$	is undefined

((44) continued)

$Intersect(\langle\{c_2, c_5\}, \{c_5\}, \{c_1, c_4\}\rangle)$	is undefined
$Intersect(\langle\{c_2, c_5\}, \{c_5\}, \{c_2\}\rangle)$	is undefined
$Intersect(\langle\{c_2, c_5\}, \{c_5\}, \{c_3, c_5\}\rangle)$	$= \{c_5\}$
$Intersect(\langle\{c_2, c_5\}, \{c_2, c_3\}, \{c_1, c_4\}\rangle)$	is undefined
$Intersect(\langle\{c_2, c_5\}, \{c_2, c_3\}, \{c_2\}\rangle)$	$= \{c_2\}$
$Intersect(\langle\{c_2, c_5\}, \{c_2, c_3\}, \{c_3, c_5\}\rangle)$	is undefined
$Intersect(\langle\{c_2, c_5\}, \{c_1, c_4\}, \{c_1, c_4\}\rangle)$	is undefined
$Intersect(\langle\{c_2, c_5\}, \{c_1, c_4\}, \{c_2\}\rangle)$	is undefined
$Intersect(\langle\{c_2, c_5\}, \{c_1, c_4\}, \{c_3, c_5\}\rangle)$	is undefined

$$\therefore K/C_{Com} = \{\{c_3\}, \{c_1, c_4\}, \{c_5\}, \{c_2\}\}$$

The last thing that still needs to be done, is to define an ordering on this new set  $K/C_{Com}$ . This needs to be done with reference to the ordering  $\leq_x$  defined in Def. 16 (38) above.

(45) **Def. 19: Second half of Step 3 in the combination process:  
the ordering  $\leq_{Com}$  on  $K/C_{Com}$ .**

Let  $(fx_{1, C1} \cap fx_{2, C2} \cap \dots \cap fx_{n, Cn}), (fy_{1, C1} \cap fy_{2, C2} \cap \dots \cap fy_{n, Cn}) \in K/C_{Com}$ .

Then  $\langle fx_{1, C1}, fx_{2, C2} \dots fx_{n, Cn} \rangle, \langle fy_{1, C1}, fy_{2, C2} \dots fy_{n, Cn} \rangle \in K/C_x$ .

Then we define the order  $\leq_{Com}$  on  $K/C_{Com}$  as follows:

$$(fx_{1, C1} \cap fx_{2, C2} \cap \dots \cap fx_{n, Cn}) \leq_{Com} (fy_{1, C1} \cap fy_{2, C2} \cap \dots \cap fy_{n, Cn}) \text{ iff}$$

$$\langle fx_{1, C1}, fx_{2, C2} \dots fx_{n, Cn} \rangle \leq_x \langle fy_{1, C1}, fy_{2, C2} \dots fy_{n, Cn} \rangle.$$

In (46) I show what the new ordered set  $\langle K/C_{Com}, \leq_{Com} \rangle$  looks like. Comparison with (33) will show that this is exactly the ordered set we were looking for.

$$(46) \quad \langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$$

$$\begin{array}{c} \{c_3\} \\ | \\ \{c_1, c_4\} \\ | \\ \{c_5\} \\ | \\ \{c_2\} \end{array}$$

The set  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  is the final output of the grammar – this is rank-ordering that EVAL imposes on the candidate set. The properties of the set  $K/C_{\text{Com}}$  and the ordering  $\leq_{\text{Com}}$  are therefore important and I will discuss them in §3.2 below. However, before we can do that, it is first necessary to prove a lemma about the mapping *Intersect*. The lemma will show that the inverse of *Intersect* is an order preserving mapping from  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  to  $\langle K/C_{\times}, \leq_{\times} \rangle$ . In and of itself this result is not of much interest. However, it will be used in §3.2 when the properties of  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  are discussed. Below I first give definitions for an inverse and an order preserving mapping, and then the lemma.

(47) **Def. 20: Inverse** (Enderton, 1977:44)

Let  $A$  and  $B$  be two sets, and  $R$  a relation on  $A \times B$ .  $R$  can then also be represented as set of ordered pairs, i.e.  $R = \{ \langle a_1, b_1 \rangle, \langle a_2, b_2 \rangle, \dots \}$  with  $a_i \in A$  and  $b_i \in B$ .

$R^{-1}: B \rightarrow A$ , the inverse of  $R$ , is then a relation on  $B \times A$ , and is defined as follows:

$$R^{-1} := \{ \langle b, a \rangle \mid \langle a, b \rangle \in R \}$$

(48) **Def. 21: The inverse of *Intersect***

Let  $K' \subseteq K$ . Then we can define  $\text{Intersect}^{-1}$  as follows:

$\text{Intersect}^{-1}$  is a relation on  $\wp(K) \times K/C_{\times}$  such that  $\text{Intersect}^{-1}(K') = \langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle$  iff  $\text{Intersect}(\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle) = K'$ .

(49) **Def. 22: An order preserving mapping** (Davey and Priestley, 1990:10)

Let  $P$  and  $Q$  be ordered sets. A map  $\varphi: P \rightarrow Q$  is said to order preserving if  $x \leq y$  in  $P$  implies  $\varphi(x) \leq \varphi(y)$  in  $Q$ .

(50) **Lemma 3: That  $Intersect^{-1}$  is order preserving**

$Intersect^{-1}$  is an order preserving mapping.

Proof of Lemma 3: Let  $Intersect(\langle fx_{1,c_1}, fx_{2,c_2} \dots fx_{n,c_n} \rangle) = K'$  and  $Intersect(\langle fy_{1,c_1}, fy_{2,c_2} \dots fy_{n,c_n} \rangle) = K''$ . Then  $K', K'' \in K/C_{Com}$  (Def. 18 (43)). For  $Intersect^{-1}$  to be order preserving, the following must therefore be true: If  $K' \leq_{Com} K''$  then  $Intersect^{-1}(K') \leq_x Intersect^{-1}(K'')$ . By substitution, this implies that for  $Intersect^{-1}$  to be order preserving,  $K' \leq_{Com} K''$  has to mean that also  $\langle fx_{1,c_1}, fx_{2,c_2} \dots fx_{n,c_n} \rangle \leq_x \langle fy_{1,c_1}, fy_{2,c_2} \dots fy_{n,c_n} \rangle$ . But this follows directly from the definition of  $\leq_{Com}$  (Def. 19 (45)).

□

Consider again the example that we have been discussing all along. In (44) I showed the result of applying  $Intersect$  to the set  $K/C_x$ . The set  $K/C_{Com}$  collects the output of  $Intersect$ .  $Intersect^{-1}$  operates on  $K/C_{Com}$ . This is shown in (51).

(51) **Applying  $Intersect^{-1}$  to  $K/C_{Com}$**

$$K/C_{Com} = \{\{c_3\}, \{c_1, c_4\}, \{c_5\}, \{c_2\}\}$$

$$Intersect^{-1}(\{c_3\}) = \langle \{c_3\}, \{c_2, c_3\}, \{c_3, c_5\} \rangle$$

$$Intersect^{-1}(\{c_1, c_4\}) = \langle \{c_1, c_4\}, \{c_1, c_4\}, \{c_1, c_4\} \rangle$$

$$Intersect^{-1}(\{c_5\}) = \langle \{c_2, c_5\}, \{c_5\}, \{c_3, c_5\} \rangle$$

$$Intersect^{-1}(\{c_2\}) = \langle \{c_2, c_5\}, \{c_2, c_3\}, \{c_2\} \rangle$$

From this example it is clear that  $Intersect^{-1}$  is indeed order preserving. To see why, consider the first two mappings from (51). We know from (46) that  $\{c_3\} \leq_{Com} \{c_1, c_4\}$ . From (39) we know that  $\langle \{c_3\}, \{c_2, c_3\}, \{c_3, c_5\} \rangle \leq_{\times} \langle \{c_1, c_4\}, \{c_1, c_4\}, \{c_1, c_4\} \rangle$ , and therefore  $Intersect^{-1}(\{c_3\}) \leq_{\times} Intersect^{-1}(\{c_1, c_4\})$ . Therefore we have  $\{c_3\} \leq_{Com} \{c_1, c_4\}$  and  $Intersect^{-1}(\{c_3\}) \leq_{\times} Intersect^{-1}(\{c_1, c_4\})$ . It can easily be checked that the same is true for all other comparisons between the mappings in (51). Now that we have shown that  $Intersect^{-1}$  is an order preserving mapping, we can begin to consider the properties of the set  $\langle K/C_{Com}, \leq_{Com} \rangle$ . This is done in the next section.<sup>17</sup>

### 3.2 The properties of $K/C_{Com}$ and $\leq_{Com}$

The ordered set  $\langle K/C_{Com}, \leq_{Com} \rangle$  is the final output of the grammar – this is the ordering that EVAL imposes on the candidate set with respect to the full constraint ranking. The properties of this set are therefore important for our understanding of what an OT grammar is. This section is devoted to identifying those properties of this set that are most directly relevant to linguistic theory. First, it is necessary to confirm that this set does indeed agree with our intuitions about what the output of grammar looks like.<sup>18</sup> In particular, we need to confirm that the ordering on this set obeys the “strictness of strict

---

<sup>17</sup> The combination process of the orderings associated with individual constraints can be achieved in another way also. We can take the Cartesian product of the quotient sets associated with the two constraints that are ranked highest, i.e.  $K/C_1 \times K/C_2$ . The lexicographic order can then be imposed on this new set. The new ordered set  $\langle K/C_1 \times K/C_2, \geq_{\times} \rangle$  can then be simplified according to the steps defined above – i.e. by applying  $Intersect$  to it, and by ordering the resulting set according to  $\leq_{Com}$ . This simplified ordered set then has exactly the same form as the quotient sets associated with individual constraints (it is a partition on  $K$ , ordered as a chain with a guaranteed minimum). This simplified set can then be combined with the quotient set associated with the third constraint in the hierarchy,  $K/C_3$ , in exactly the same manner as described for  $K/C_1$  and  $K/C_2$ . This process is repeated recursively until the quotient sets associated with all of the constraints have been incorporated. The result of this process is provably equivalent to the process as I have defined it in the text.

<sup>18</sup> See the discussion on the introductory section to this chapter. We need to confirm that the explicatum agrees with our intuitions about the explicandum.

domination” principle. This is the focus of §3.2.1. After this some properties of the set  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  are discussed. Section §3.2.2 first shows that that  $\leq_{\text{Com}}$  imposes a chain ordering on  $K/C_{\text{Com}}$ , and §3.2.3 then that this chain is guaranteed to have a minimum. Section §3.2.4 then shows that the set  $K/C_{\text{Com}}$  is a partition on the candidate set. Finally, §3.2.5 shows that the members of  $K/C_{\text{Com}}$  are equivalence classes on the candidate set  $K$ .

### 3.2.1 $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$ and the strictness of strict domination

An OT grammar is a ranking between the constraints in CON. The candidate set is ordered relative to the constraint ranking with higher ranked constraints taking absolute precedence over lower ranked constraints. In particular, the highest ranked constraint that differentiates between two candidates takes precedence over all lower ranked constraints that also differentiate between these two candidates. Concretely, consider two candidates  $k_1$  and  $k_2$ , and two constraints  $C_1$  and  $C_2$  ranked as  $\|C_1 \circ C_2\|$ . Suppose that  $C_1$  prefers  $k_1$  over  $k_2$ , but that  $C_2$  prefers  $k_2$  over  $k_1$ , i.e.  $C_1(k_1) < C_1(k_2)$  but  $C_2(k_1) > C_2(k_2)$ . Because of the ranking  $\|C_1 \circ C_2\|$ , the ordering that  $C_1$  imposes on  $k_1$  and  $k_2$  takes precedence – that is, for this mini-grammar as a whole,  $k_1$  is more harmonic than  $k_2$ . This property of an OT grammar has been dubbed the “strictness of strict domination” or “strictness of domination” (McCarthy, 2002b:4, Prince and Smolensky, 1993:78, 1997:1604).

Since classic OT typically cares only about the relation between the winner and the mass of losers, strictness of domination is usually discussed as if it applies only to the relations between the winner and the losers. However, there is nothing that prohibits us from assuming that it also applies to the relationships between the losers, and that is in fact the assumption that I make in this dissertation. Strictness of domination therefore applies to the harmonic ordering relationships between any two candidates. In general



then, it is the highest ranked constraint in terms of which any two candidates are differentiated that determines the harmonic relation between them. For any two candidates  $k_1$  and  $k_2$ , if  $C_1$  is the highest ranked constraint that differentiates between them, then these two candidates will be harmonically ordered in terms of  $C_1$ . Stated differently, if  $C_1(k_1) < C_1(k_2)$ , then  $|k_1 \text{ }^{\text{TM}} k_2|$  even if there is some lower ranked constraint  $C_2$  such that  $C_2(k_1) > C_2(k_2)$ .

The set  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  can only be viewed as the output of an OT grammar, if it can be shown that this set abides by the strictness of domination principle. This requirement is stated somewhat informally in (52).

(52) **Strictness of domination with reference to  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$ , somewhat informally**

Let  $K_1, K_2 \in K/C_{\text{Com}}$ , with  $k_1 \in K_1$  and  $k_2 \in K_2$ . Then:

$K_1 <_{\text{Com}} K_2$  iff the highest ranked constraint that distinguishes between  $k_1$  and  $k_2$  prefers  $k_1$  over  $k_2$ .

The rest of this section is dedicated to showing that  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  does indeed abide by the strictness of domination principle. This is done in three steps: First, the concept “crucial constraint” is defined. The crucial constraint for two candidates is the highest ranked constraint that differentiates between them. Then, strictness of domination with reference to  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  is defined more precisely by using the concept of crucial constraints. Finally, it is proved that  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  abides by strictness of domination.

(53) **Def. 23: Crucial constraints**

Let  $k_1, k_2 \in K$ , and let the grammar under consideration be  $\|C_1 \circ C_2 \circ \dots \circ C_n\|$ .

Then we define  $Crux_{1,2}$ , the crucial constraint for  $k_1$  and  $k_2$ , as follows:

$$Crux_{1,2} = C_i \text{ such that } C_i(k_1) \neq C_i(k_2) \text{ and } \neg \exists j (j < i \text{ and } C_j(k_1) \neq C_j(k_2)).^{19}$$

This definition identifies for any two candidates the highest ranked constraint that assigns a different number of violations to these two candidates. Now that the concept of a crucial constraint has been defined, we can state more formally what it would mean for the set  $\langle K/C_{Com}, \leq_{Com} \rangle$  to abide by the strictness of domination principle. In (54) I first state the formal definition of strictness of domination. In (55) I then prove that  $\langle K/C_{Com}, \leq_{Com} \rangle$  does indeed abide by this principle.

(54) **Def. 24: Strictness of domination with reference to  $\langle K/C_{Com}, \leq_{Com} \rangle$**

Let  $k_1, k_2 \in K$ , and  $K_1, K_2 \in K/C_{Com}$  such that  $k_1 \in K_1$  and  $k_2 \in K_2$ .

Let  $fk_{i,C_j}$  be the equivalence class of  $k_i \in K$  in terms of constraint  $C_j$  as defined in Def. 4 (13) above, and  $\leq_{C_j}$  the ordering associated with this constraint as defined in Def. 6 (17) above.

Let  $Crux_{1,2}$  be the crucial constraint as defined just above in Def. 23 (53), and  $\leq_{Crux_{1,2}}$  the ordering that EVAL imposes on the candidate set with reference to  $Crux_{1,2}$ . Then:

$$K_1 <_{Com} K_2 \text{ iff } fk_{1,Crux_{1,2}} <_{Crux_{1,2}} fk_{2,Crux_{1,2}}.$$

---

<sup>19</sup> Note that this definition implies that there will be no crucial constraints for two candidates that receive the same number of violations in terms of every constraint. This does not negate the result proved just below about the strictness of strict domination. Since all constraints agree on the relationship between two such candidates, it is not possible that a later constraint can disagree on an earlier constraint on the ordering relationship between two such candidates.

(55) **Theorem 7: Strictness of domination and  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$**

$\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  abides by strictness of domination.

Proof of Theorem 7: Let  $k_1, k_2 \in K$ , and  $K_1, K_2 \in K/C_{\text{Com}}$  such that  $k_1 \in K_1$  and  $k_2 \in K_2$ . Let  $\text{Crux}_{1,2}$  be the crucial constraint for  $k_1$  and  $k_2$  as defined in Def. 23 (53), and  $\leq_{\text{Crux}_{1,2}}$  the ordering associated with this constraint as defined in Def. 6 (17).

We need to show that  $K_1 <_{\text{Com}} K_2$  iff  $f_{k_1, \text{Crux}_{1,2}} <_{\text{Crux}_{1,2}} f_{k_2, \text{Crux}_{1,2}}$ . I will start by showing that  $K_1 <_{\text{Com}} K_2$  implies  $f_{k_1, \text{Crux}_{1,2}} <_{\text{Crux}_{1,2}} f_{k_2, \text{Crux}_{1,2}}$ . The basic strategy in this proof is to work backwards through successive definitions until a statement can be made in terms of constraints. Since constraints have  $\cup$  as range, it is easy to draw inferences on orderings associated with constraints.

Assume  $K_1 <_{\text{Com}} K_2$ . By Def. 18 (43) and Def. 17 (42), the existence of  $K_1, K_2 \in K/C_{\text{Com}}$  implies the existence of some  $\langle f_{x_1, C_1}, f_{x_2, C_2} \dots f_{x_n, C_n} \rangle \in K/C_{\times}$  such that  $K_1 = (f_{x_1, C_1} \cap f_{x_2, C_2} \cap \dots \cap f_{x_n, C_n})$ , and similarly the existence of some  $\langle f_{y_1, C_1}, f_{y_2, C_2} \dots f_{y_n, C_n} \rangle \in K/C_{\times}$  such that  $K_2 = (f_{y_1, C_1} \cap f_{y_2, C_2} \cap \dots \cap f_{y_n, C_n})$ . From this it also follows that  $\text{Intesect}^{-1}(K_1) = \langle f_{x_1, C_1}, f_{x_2, C_2} \dots f_{x_n, C_n} \rangle$ , and  $\text{Intesect}^{-1}(K_2) = \langle f_{y_1, C_1}, f_{y_2, C_2} \dots f_{y_n, C_n} \rangle$  (Def. 21 (48)). And since  $\text{Intersect}^{-1}$  is order preserving (Lemma 3 (50)),  $K_1 <_{\text{Com}} K_2$  implies  $\langle f_{x_1, C_1}, f_{x_2, C_2} \dots f_{x_n, C_n} \rangle <_{\times} \langle f_{y_1, C_1}, f_{y_2, C_2} \dots f_{y_n, C_n} \rangle$ . By the definition of  $\leq_{\times}$  (Def. 16 (38)), we then have that there is some  $i$  such that for all  $j \leq i$ ,  $f_{x_j, C_j} =_{C_j} f_{y_j, C_j}$  and  $f_{x_i, C_i} <_{C_i} f_{y_i, C_i}$ . Then by the definition of  $\leq_{C_i}$  (Def. 6 (17)), it follows that there is some  $i$  such that for all  $j \leq i$ ,  $C_j(x_j) = C_j(y_j)$  and  $C_i(x_i) < C_i(y_i)$ .  $C_i$  is then the highest ranked constraint in terms of which the candidates in  $K_1$  and  $K_2$  differ.

Since  $k_1 \in K_1$  and  $k_2 \in K_2$  (by assumption), we therefore also have  $C_i(k_1) < C_i(k_2)$ . But since  $C_i$  is also the highest ranked constraint in terms of which  $k_1$  and  $k_2$  differ, it follows by Def. 23 (53) of crucial constraints that  $C_i$  is the crucial constraint for  $k_1$  and  $k_2$ , i.e.  $C_i = Crux_{1,2}$ . Therefore we have  $Crux_{1,2}(k_1) < Crux_{1,2}(k_2)$ . But by the definition of the ordering  $\leq_C$  (Def. 6 (17)) we then have  $f_{k_1, Crux_{1,2}} <_{Crux_{1,2}} f_{k_2, Crux_{1,2}}$ . And therefore we have that  $K_1 <_{Com} K_2$  implies  $f_{k_1, Crux_{1,2}} <_{Crux_{1,2}} f_{k_2, Crux_{1,2}}$ .

And the argument can be reversed to prove the converse. □

We can again check to confirm that this Theorem is indeed true of our example. In (56) I repeat from (39) the orderings  $\leq_{C_i}$  associated with each of the three constraints, as well as the ordering  $\leq_{Com}$  from (46) associated with the grammar as a whole.

(56) **Orderings associated with constraints and the grammar as a whole**

$\langle K/C_1, \leq_{C_1} \rangle$	$\langle K/C_2, \leq_{C_2} \rangle$	$\langle K/C_3, \leq_{C_3} \rangle$	$\langle K/C_{Com}, \leq_{Com} \rangle$
$\{c_3\}$	$\{c_5\}$	$\{c_1, c_4\}$	$\{c_3\}$
$\{c_1, c_4\}$	$\{c_2, c_3\}$	$\{c_2\}$	$\{c_1, c_4\}$
$\{c_2, c_5\}$	$\{c_1, c_4\}$	$\{c_3, c_5\}$	$\{c_5\}$
			$\{c_2\}$

What we see is that the final ordering  $\leq_{Com}$  does not contradict the ordering  $\leq_{C_1}$  associated with the highest ranked constraint  $C_1$ . For instance, we have  $\{c_3\} <_{C_1} \{c_2, c_5\}$  which implies that  $C_1(c_3) < C_1(c_5)$ . On the other hand we have  $\{c_5\} <_{C_2} \{c_2, c_3\}$  which implies  $C_2(c_5) < C_2(c_3)$ .  $C_1$  and  $C_2$  therefore conflict in how they rate  $c_3$  and  $c_5$ . However, since  $C_1$  dominates  $C_2$ , the ordering  $\leq_{Com}$  associated with the grammar as a whole agrees with the ordering  $\leq_{C_1}$  associated with  $C_1$  – i.e.  $\{c_3\} <_{Com} \{c_5\}$ .

We have therefore now established that  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  abides by strictness of domination. This means that this set does indeed agree with our intuitions about what the output of an OT grammar should be like. This confirms that the procedures described above for arriving at this set, is an accurate depiction of what EVAL does to the candidate set. Now that we have established that the set  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  is of the correct form, we can investigate some its properties in more detail.

### 3.2.2 $\leq_{\text{Com}}$ defines a chain

In §2.2.2 I showed that the ordering that EVAL imposes on the candidate set with regard to each individual constraint is a chain ordering. In this section I will show that this also true of the ordering that EVAL imposes on the candidate set in terms of the grammar as a whole – i.e. not only is  $\leq_C$  for every  $C \in \text{CON}$  a chain, but so is the result of combining each of these orderings into the conglomerate ordering  $\leq_{\text{Com}}$ . For a discussion of what a chain see, refer to Def. 10 (26) above.

#### (57) **Theorem 8: That $\leq_{\text{Com}}$ defines a chain**

The ordering that  $\leq_{\text{Com}}$  imposes on  $K/C_{\text{Com}}$  is a chain.

Proof of Theorem 8: The proof presented here follows the same basic strategy as the proof above for Theorem 7 – that is, substituting backwards through successive definitions. Let  $K_1, K_2 \in K/C_{\text{Com}}$ , with  $K_1$  and  $K_2$  not necessarily distinct.

By Def. 18 (43) and Def. 17 (42), the existence of  $K_1, K_2 \in K/C_{\text{Com}}$  implies the existence of some  $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle \in K/C_{\times}$  such that  $K_1 = (fx_{1,C1} \cap fx_{2,C2} \cap \dots \cap fx_{n,Cn})$ , and similarly the existence of some  $\langle fy_{1,C1}, fy_{2,C2} \dots fy_{n,Cn} \rangle \in K/C_{\times}$  such that  $K_2 = (fy_{1,C1} \cap fy_{2,C2} \cap \dots \cap fy_{n,Cn})$ .

Now, let  $\leq_{C_i}$  be the ordering that EVAL imposes on the candidate set relative to constraint  $C_i$  (Def. 6 (17)). Since  $\leq_{C_i}$  is a chain (Theorem 4 (27)), there are now three possible scenarios:

Scenario 1: For all  $i \leq n$ ,  $fx_{i,C_i} =_{C_i} fy_{i,C_i}$ . Then, by the definition of  $\leq_x$  (Def. 16 (38)), it follows that  $\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle =_x \langle fy_{1,C_1}, fy_{2,C_2} \dots fy_{n,C_n} \rangle$ . This again implies by the definition of  $\leq_{\text{Com}}$  (Def. 19 (45)), that  $(fx_{1,C_1} \cap fx_{2,C_2} \cap \dots \cap fx_{n,C_n}) =_{\text{Com}} (fy_{1,C_1} \cap fy_{2,C_2} \cap \dots \cap fy_{n,C_n})$ , and therefore that  $K_1 =_{\text{Com}} K_2$ .

Scenario 2: There is some  $k$  such that  $fx_{k,C_k} >_{C_k} fy_{k,C_k}$ , and for all  $i \leq k$ ,  $fx_{i,C_i} =_{C_i} fy_{i,C_i}$ . Then, by the definition of  $\leq_x$  (Def. 16 (38)), it follows that  $\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle >_x \langle fy_{1,C_1}, fy_{2,C_2} \dots fy_{n,C_n} \rangle$ , which again implies that  $(fx_{1,C_1} \cap fx_{2,C_2} \cap \dots \cap fx_{n,C_n}) >_{\text{Com}} (fy_{1,C_1} \cap fy_{2,C_2} \cap \dots \cap fy_{n,C_n})$  (Def. 19 (45)), and therefore that  $K_1 >_{\text{Com}} K_2$ .

Scenario 3: There is some  $k$  such that  $fx_{k,C_k} <_{C_k} fy_{k,C_k}$ , and for all  $i \leq k$ ,  $fx_{i,C_i} =_{C_i} fy_{i,C_i}$ . Then, by the definition of  $\leq_x$  (Def. 16 (38)), it follows that  $\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle <_x \langle fy_{1,C_1}, fy_{2,C_2} \dots fy_{n,C_n} \rangle$ , which again implies that  $(fx_{1,C_1} \cap fx_{2,C_2} \cap \dots \cap fx_{n,C_n}) <_{\text{Com}} (fy_{1,C_1} \cap fy_{2,C_2} \cap \dots \cap fy_{n,C_n})$  (Def. 19 (45)), and therefore that  $K_1 <_{\text{Com}} K_2$ .

Finally, we then have that either  $K_1 =_{\text{Com}} K_2$ , or  $K_1 >_{\text{Com}} K_2$ , or  $K_1 <_{\text{Com}} K_2$ , and therefore that  $\leq_{\text{Com}}$  defines a chain on  $K/C_{\text{Com}}$ . □

It is clear that this Theorem is true of the example that we have been using throughout the discussion. Referring back to the ordering  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  in (56) confirms that indeed any two elements on this ordering are comparable.

Recall that the set  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  is the final output of the grammar. This set represents the ordering that EVAL imposes on the candidate set with reference to the complete constraint hierarchy. The fact that this set is ordered as a chain is important. It shows that there is no indeterminacy in the output of the grammar. For any given candidate, it is possible to determine how it is harmonically related to any other candidate. Consider any two candidates  $k_1$  and  $k_2$ . If  $k_1$  and  $k_2$  are both members of the same member of  $K/C_{\text{Com}}$ , i.e.  $k_1, k_2 \in K_1 \in K/C_{\text{Com}}$ , then  $k_1$  and  $k_2$  are equally harmonic. The grammar can then not distinguish between these two candidates, and they can be considered as grammatically indistinct.<sup>20</sup> However, if  $k_1$  and  $k_2$  belong to different members,  $K_1$  and  $K_2$ , of  $K/C_{\text{Com}}$ , i.e.  $k_1 \in K_1, k_2 \in K_2$  and  $K_1 \neq K_2$ , then either  $k_1$  is more harmonic than  $k_2$  or  $k_2$  is more harmonic than  $k_1$ .

### 3.2.3 The chain defined by $\leq_{\text{Com}}$ always has a minimum

In §2.2.3 above I showed that the chain ordering  $\leq_C$  associated with each individual constraint  $C$  is guaranteed to have a minimum. In this section I will show that the same is true for the ordering  $\leq_{\text{Com}}$  (the ordering that EVAL imposes on the candidate set with reference to the whole grammar). For a definition of a minimum, see Def. 11 (28) above.

(58) **Theorem 9: That  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  always has a minimum**

The ordering  $\leq_{\text{Com}}$  always has a minimum in  $K/C_{\text{Com}}$ .

Proof of Theorem 9: This theorem is proved by assuming the opposite, and then showing that this leads to a contradiction. In particular, I will show that assuming that

---

<sup>20</sup> See the discussion in §3.2.5 below about grammatical indistinctness.

$\leq_{\text{Com}}$  does not have a minimum in  $K/C_{\text{Com}}$  implies that the ordering  $\leq_x$  on  $K/C_x$  does not have a minimum, contra Lemma 2 (41).

Assume that  $\leq_{\text{Com}}$  does not have a minimum in  $K/C_{\text{Com}}$ . We have established just above that  $\leq_{\text{Com}}$  defines a chain on  $K/C_{\text{Com}}$  (Theorem 8 (57)). From this it follows that the only way in which  $\leq_{\text{Com}}$  cannot have a minimum in  $K/C_{\text{Com}}$ , is if  $\leq_{\text{Com}}$  defines an infinitely descending chain on  $K/C_{\text{Com}}$ . But if  $\leq_{\text{Com}}$  defines an infinitely descending chain on  $K/C_{\text{Com}}$ , then for all  $K_1 \in K/C_{\text{Com}}$ , we have that there is some  $K_2 \in K/C_{\text{Com}}$  such that  $K_2 <_{\text{Com}} K_1$ .

We have seen above that *Intersect*<sup>1</sup> is an order preserving map between  $K/C_{\text{Com}}$  and  $K/C_x$  (Lemma 3 (50)). The ordering  $\leq_x$  on  $K/C_x$  will therefore inherit the properties of the ordering  $\leq_{\text{Com}}$  on  $K/C_{\text{Com}}$ . It then follows that  $\leq_x$  on  $K/C_x$  is also an infinitely descending chain. But this contradicts Lemma 2 (41) which asserts that  $\leq_x$  is guaranteed to have a minimum in  $K/C_x$ . □

The dual of this theorem is obviously not true – that is, it is not the case that the set  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  is guaranteed to have a maximum. There are constraints that can in principle assign an unbounded number of violations (such as DEP, ONSET, etc.).<sup>21</sup> Because of this, it is possible that  $\leq_{\text{Com}}$  can define an infinitely ascending chain on  $K/C_{\text{Com}}$ .

---

<sup>21</sup> See footnote 10 above.



It is again clear that this Theorem is true of the example that we have been discussing throughout.  $\{c_3\}$  in (56) precedes all other members of  $K/C_{Com}$  in terms of the ordering  $\leq_{Com}$ , and  $\{c_3\}$  is therefore the minimum of the ordering  $\leq_{Com}$  in the set  $K/C_{Com}$ .

Theorem 9 is a very important result. The set  $\langle K/C_{Com}, \leq_{Com} \rangle$  is the final output of the grammar. In order for this output to be in agreement with the way that we standardly think about an OT grammar, it should be possible to read off this set what the optimal candidate of classic OT is. This optimal candidate is the minimum of the set  $K/C_{Com}$  under the ordering  $\leq_{Com}$ . Since this minimum is guaranteed to exist it follows that we will always be able to determine what the optimal candidate of classic OT should be.

But there is another reason why it is important that the set  $\langle K/C_{Com}, \leq_{Com} \rangle$  is guaranteed to have a minimum. From Theorem 8 (57) we know that  $\langle K/C_{Com}, \leq_{Com} \rangle$  is a chain. If this chain had no minimum and no maximum, then it would have been impossible to identify individual members of this set with reference to the ordering on the set. For any member of this set there would then always have been another member that precedes it and another member that follows it. How would language users then access the information contained in this ordered set? There has to be a unique point on the ordering with regard to which the members of the set can be identified in terms of the ordering on the set. The minimum on the chain can serve this purpose. Since only one member of the set  $K/C_{Com}$  can be the minimum of this set under the ordering  $\leq_{Com}$ , we can uniquely identify this one member of  $K/C_{Com}$  in terms of the ordering  $\leq_{Com}$ . All other members of  $K/C_{Com}$  can then also be uniquely identified by stating their relationship to the minimum in terms  $\leq_{Com}$ . This minimum on the set  $\langle K/C_{Com}, \leq_{Com} \rangle$  serves as the point

through which language users can access the information contained in the set  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$ .

### 3.2.4 The set $K/C_{\text{Com}}$ is a partition on $K$

In §2.3.4 above I showed the quotient set  $K/C$  associated with every constraint  $C$  is a partition on the candidate set  $K$ . This means that every candidate  $k \in K$  was contained in exactly one member of  $K/C$ . This result was important. The fact that every candidate is contained in a member of  $K/C$  implies that the ordered set  $\langle K/C, \leq_C \rangle$  contains information about every candidate. The fact that every candidate is contained in only one member of  $K/C$  implies that every candidate can occur in only one place in the ordering  $\leq_C$ . The final output of the grammar is the set  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$ . In this section I will show that this set is also a partition on the candidate set  $K$ . This result will assure that also in this final output of the grammar every candidate is guaranteed to occupy exactly one position. For a definition of a partition, refer to Def. 12 (30) above.

#### (59) **Theorem 10: That $K/C_{\text{Com}}$ is a partition on $K$**

$K/C_{\text{Com}}$  is a partition on  $K$ .

Proof of Theorem 10: I will consider each of the three requirements for  $K/C_{\text{Com}}$  to be a partition on  $K$  in turn.

Consider first the requirement that  $K/C_{\text{Com}}$  consists of non-empty subsets of  $K$ . This follows directly from the definitions of  $K/C_{\text{Com}}$  (Def. 18 (43)) and *Intersect* (Def. 17 (42)).  $K/C_{\text{Com}}$  is defined such that all of its members are in the image of the set  $K/C_{\times}$

under the relation *Intersect*.<sup>22</sup> And *Intersect* is defined as relation into  $\wp(K)$  and such that  $\emptyset$  is not in its range. We therefore have that all the members of  $K/C_{\text{Com}}$  are non empty subsets of  $K$ .

Now consider the second requirement, that the members  $K/C_{\text{Com}}$  be exhaustive subsets of  $K$ , i.e. that every  $k \in K$  is a member of some member of  $K/C_{\text{Com}}$ . For all  $C \in \text{CON}$ , let  $K/C$  be the quotient set associated with constraint  $C$  as defined in Def. 5 (15), and  $fx_{,C} \in K/C$ , the equivalence class of  $x$  in terms of  $C$  (Def. 4 (13)). By Theorem 6 (31) we then have that  $K/C$  is a partition on  $K$ , and therefore for every  $k \in K$  there is some  $fx_{,C} \in K/C$  such that  $k \in fx_{,C}$ .

Now, let  $K/C_{\times}$  be the Cartesian product over the quotient sets associated with each constraint (Def. 14 (35)), and  $\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle \in K/C_{\times}$ . Then for every  $k \in K$  there is some  $\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle \in K/C_{\times}$  such that  $k \in fx_{i,C_i}$  for all  $i \leq n$ . From the definition of *Intersect* (Def. 17 (42)), we then have that for every  $k \in K$ , there is some  $\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle \in K/C_{\times}$  such that  $k \in \text{Intersect}(\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle)$ . And from the definition of  $K/C_{\text{Com}}$  (Def. 18 (43)) we have that  $\text{Intersect}(\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle) \in K/C_{\text{Com}}$ . Therefore, we have for every  $k \in K$  that  $k \in \text{Intersect}(\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle) \in K/C_{\text{Com}}$ .

Now consider the third requirement, that the members of  $K/C_{\text{Com}}$  be disjoint subsets of  $K$ . I will show that any two members of  $K/C_{\text{Com}}$  that have an element in common are identical. Let  $K_1, K_2 \in K/C_{\text{Com}}$  such  $k \in K_1$  and  $k \in K_2$ . The existence of  $K_1$ ,

---

<sup>22</sup> The image of set  $A$  under the relation  $F$  is defined as the set of all  $u$  such that there is some  $v \in A$  such that  $u = F(v)$  (Enderton, 1977:44), i.e. the image of  $A$  under  $F := \{v \mid \exists u (u \in A \ \& \ F(u) = v)\}$ .

$K_2$  implies the existence of  $\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle \in K/C_\times$  and  $\langle fy_{1,C_1}, fy_{2,C_2} \dots fy_{n,C_n} \rangle \in K/C_\times$  such that  $Intersect(\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle) = K_1$  and  $Intersect(\langle fy_{1,C_1}, fy_{2,C_2} \dots fy_{n,C_n} \rangle) = K_2$  (Def. 17 (42) and Def. 18 (43)). Therefore  $K_1 = fx_{1,C_1} \cap fx_{2,C_2} \cap \dots \cap fx_{n,C_n}$ , and  $K_2 = fy_{1,C_1} \cap fy_{2,C_2} \cap \dots \cap fy_{n,C_n}$ . But since  $k \in K_1$  and  $k \in K_2$  (by assumption), it follows that  $k \in fx_{1,C_1} \cap fx_{2,C_2} \cap \dots \cap fx_{n,C_n}$  and  $k \in fy_{1,C_1} \cap fy_{2,C_2} \cap \dots \cap fy_{n,C_n}$ . Therefore, for all  $i \leq n$ ,  $k \in fx_{i,C_i}$  and  $k \in fy_{i,C_i}$ .

But if  $k \in fx_{i,C_i}$ , then  $x_i \approx_{C_i} k$ , and similarly if  $k \in fy_{i,C_i}$ , then  $y_i \approx_{C_i} k$  (Def. 4 (13)). But  $\approx_{C_i}$  is an equivalence relation, and therefore symmetric and transitive (Theorem 2 (12)). Therefore, if  $y_i \approx_{C_i} k$ , then  $k \approx_{C_i} y_i$ , and if  $(x_i \approx_{C_i} k$  and  $k \approx_{C_i} y_i)$ , then  $x_i \approx_{C_i} y_i$ . From this it follows that  $x_i \in fy_{i,C_i}$ , so that  $(x_i \in fy_{i,C_i}$  and  $x_i \in fx_{i,C_i})$ . But  $fx_{i,C_i}$  and  $fy_{i,C_i}$  are both elements of  $K/C_i$ , the quotient set associated with  $C_i$  (Def. 5 (15)). And we know that  $K/C_i$  is a partition on  $K$  (Theorem 6 (31)). The members of  $K/C_i$  are therefore disjoint, and therefore  $(x_i \in fy_{i,C_i}$  and  $x_i \in fx_{i,C_i})$  implies  $fy_{i,C_i} = fx_{i,C_i}$ . Then we have that  $\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle = \langle fy_{1,C_1}, fy_{2,C_2} \dots fy_{n,C_n} \rangle$ .

From  $\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle = \langle fy_{1,C_1}, fy_{2,C_2} \dots fy_{n,C_n} \rangle$  it follows that  $Intersect(\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle) = Intersect(\langle fy_{1,C_1}, fy_{2,C_2} \dots fy_{n,C_n} \rangle)$ . And since  $K_1 = Intersect(\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle)$  and  $K_2 = Intersect(\langle fy_{1,C_1}, fy_{2,C_2} \dots fy_{n,C_n} \rangle)$ , it follows that  $K_1 = K_2$ .

Finally, we then have that  $k \in K_1$  and  $k \in K_2$  implies  $K_1 = K_2$ , and therefore that the members of  $K/C_{Com}$  are disjoint. □

We can again check that this Theorem holds of the mini grammar that have been considering as an example throughout this chapter. The set  $K/C_{\text{Com}}$  for our mini grammar is  $\{\{c_3\}, \{c_1, c_4\}, \{c_5\}, \{c_2\}\}$  (see (51)). In our example the candidate set  $K$  has only five members, i.e.  $K = \{c_1, c_2, c_3, c_4, c_5\}$ . It is clear that each member of  $K$  is included in exactly one member of  $K/C_{\text{Com}}$ .

The fact that  $K/C_{\text{Com}}$  is a partition on  $K$  is an important result. The ordered set  $\langle K/C_{\text{Com}}, \leq_{\text{Com}} \rangle$  is the final output of the grammar. The fact that every candidate is included in one member of  $K/C_{\text{Com}}$  implies that every candidate is represented in the final output of the grammar. The fact that every candidate is included in only member of  $K/C_{\text{Com}}$  implies that every candidate occupies a unique place in the final output of the grammar. Together with the fact that  $\leq_{\text{Com}}$  defines a chain on  $K/C_{\text{Com}}$  (Theorem 8 (57)), this means that there is no indeterminacy in the output of the grammar. It is always possible for any two candidates to determine how they are harmonically related to each other with regard to the ranking between the constraints. Consider  $K_1, K_2 \in K/C_{\text{Com}}$ , and  $k_1 \in K_1, k_2 \in K_2$ . If  $K_1 =_{\text{Com}} K_2$ , then we know that  $k_1$  and  $k_2$  are equally well-formed in terms of the grammar under consideration. If  $K_1 <_{\text{Com}} K_2$ , then we know that  $k_1$  is more well-formed than  $k_2$ . If  $K_1 >_{\text{Com}} K_2$ , then we know that  $k_1$  is less well-formed than  $k_2$ . Of particular relevance for the topic of this dissertation, is that this implies that any two candidates can be ordered with respect to each other – not only the best candidate in relation to the losers, but even any two losers in relation to each other.

### 3.2.5 The members of $K/C_{\text{Com}}$ as equivalence classes on $K$

In §2 above I discussed the quotient sets  $K/C_i$  associated with the different constraints  $C_i$ . These quotient sets are not sets of candidates, but sets of sets of candidates. The set

$K/C_{\text{Com}}$  is similar to these quotient sets. It is also a set of sets of candidates. However, there is an important difference between  $K/C_{\text{Com}}$  and the sets associated with individual constraints. Let  $K/C_1$  be the quotient set associated with constraint  $C_1$ , and  $f_{k_1, C_1} \in K/C_1$ .  $f_{k_1, C_1}$  is therefore an equivalence class in terms of constraint  $C_1$ , implying that all candidates in  $f_{k_1, C_1}$  receive the same number of violations in terms of  $C_1$ . If we have both  $k_1, k_2 \in f_{k_1, C_1}$ , then  $C_1(k_1) = C_1(k_2)$ . In terms  $C_1$  these two candidates are indistinguishable. However it is possible that there is some other constraint in terms of which these two candidates differ – i.e. there could be some other constraint  $C_2$  such that  $C_2(k_1) \neq C_2(k_2)$ . Although  $k_1$  and  $k_2$  are indistinguishable in terms of  $C_1$  they might still be distinct from each other.

Now consider the set  $K' \in K/C_{\text{Com}}$ , and  $k_1, k_2 \in K'$ . The two candidates  $k_1$  and  $k_2$  are now completely indistinguishable in terms of the grammar – they receive exactly the same number violations in terms of every constraint. Two candidates that belong to same set of candidates in  $K/C_{\text{Com}}$  are therefore grammatically indistinct.<sup>23</sup> Although each member of  $K/C_{\text{Com}}$  can therefore contain more than one candidate, the grammar cannot distinguish between them. When we talk about a “candidate” in the output of an OT grammar, what we are actually referring to is rather a set of grammatically indistinct candidates. In this section I will prove that the sets in  $K/C_{\text{Com}}$  do indeed contain only grammatically indistinct candidates. In order to show this, I will first define an equivalence relation  $\approx_{\text{Com}}$  on the candidate set, and then show that the members of  $K/C_{\text{Com}}$

---

<sup>23</sup> Samek-Lodovici and Prince (1999) also use the concept of grammatical indistinctness. See also Hammond (1994, 2000) who uses this idea as a method of accounting for variation in the output – he assumes that variation arises when the set of best candidates has more than one member.

can be defined in terms of  $\approx_{\text{Com}}$ . For a definition of an equivalence relation, see Def. 3 (11) above.

(60) **Def. 25:  $\approx_{\text{Com}}$  as a relation on  $K$**

Let the grammar under consideration be  $\|C_1 \circ C_2 \circ \dots \circ C_n\|$ , and  $k_1, k_2 \in K$ .

Then we define  $\approx_{\text{Com}}$  as follows:

$$\approx_{\text{Com}} \subseteq K \times K \text{ such that } k_1 \approx_{\text{Com}} k_2 \text{ iff for all } i \leq n \ C_i(k_1) = C_i(k_2)$$

(61) **Theorem 11: That  $\approx_{\text{Com}}$  is an equivalence relation**

$\approx_{\text{Com}}$  is an equivalence relation on  $K$ .

Proof of Theorem 11:  $\approx_{\text{Com}}$  is by definition a binary relation on  $K$ . All that needs to be shown then is that it is reflexive, transitive and symmetric. Since it is defined in terms of constraints this is a rather straightforward matter. Constraints are functions into  $\mathbb{U}$ , and the relation  $=$  on  $\mathbb{U}$  is reflexive, transitive and symmetric. I consider each of the three requirements in turn below.

(i) That  $\approx_{\text{Com}}$  is reflexive. Consider any candidate  $k \in K$ . Since  $C(k) = C(k)$  for all  $C \in \text{CON}$ , it follows from the definition of  $\approx_{\text{Com}}$  (Def. 25 (60)) that  $k \approx_{\text{Com}} k$ . Therefore,  $\approx_{\text{Com}}$  is reflexive.

(ii) That  $\approx_{\text{Com}}$  is transitive. Consider any three candidates  $k_1, k_2, k_3 \in K$  such that  $k_1 \approx_{\text{Com}} k_2$  and  $k_2 \approx_{\text{Com}} k_3$ . By the definition of  $\approx_{\text{Com}}$  (Def. 25 (60)) we then have that for all constraints  $C \in \text{CON}$ ,  $C(k_1) = C(k_2)$  and  $C(k_2) = C(k_3)$ . But since  $=$  is transitive on  $\mathbb{U}$ , it follows from that  $C(k_1) = C(k_3)$ . From the definition of  $\approx_{\text{Com}}$  it then follows that  $k_1 \approx_{\text{Com}} k_3$ . Therefore,  $\approx_{\text{Com}}$  is transitive.

(iii) That  $\approx_{\text{Com}}$  is symmetric. Consider any two candidates  $k_1, k_2 \in K$  such that  $k_1 \approx_{\text{Com}} k_2$ . By the definition of  $\approx_{\text{Com}}$  (Def. 25 (60)) we then have that for all constraints  $C \in \text{CON}$ ,  $C(k_1) = C(k_2)$ . But since  $=$  is symmetric on  $\dot{\cup}$ , it follows that  $C(k_1) = C(k_2)$  implies  $C(k_2) = C(k_1)$ . And then again by the definition of  $\approx_{\text{Com}}$  it follows that  $k_2 \approx_{\text{Com}} k_1$ . Therefore,  $\approx_{\text{Com}}$  is symmetric.  $\square$

Now we can show that the members of the set  $K/C_{\text{Com}}$  can be fully defined in terms of the equivalence relation  $\approx_{\text{Com}}$ .

(62) **Theorem 12: The members of  $K/C_{\text{Com}}$  can be fully defined in terms of  $\approx_{\text{Com}}$**

Let  $K$  be the candidate set,  $K_1 \in K/C_{\text{Com}}$  and  $k_1 \in K_1$ . Then:

- (i)  $\forall k_2 \in K_1, k_1 \approx_{\text{Com}} k_2$ , and
- (ii)  $\forall k_3 \in K$  such that  $k_1 \approx_{\text{Com}} k_3, k_3 \in K_1$ .

Proof of Theorem 12: Let  $fk_{i,C_j}$  be the equivalence class of candidate  $k_i$  in terms of constraint  $C_j$  (Def. 4 (13)). Now we can consider each of the two clauses of Theorem 12 in turn.

(i) We have by assumption that  $k_1, k_2 \in K_1$ . And the existence of  $K_1$  implies that there is some  $n$ -tuple  $\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle \in K/C_{\times}$  such that  $K_1 = \text{Intersect}(\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle) = fx_{1,C_1} \cap fx_{2,C_2} \cap \dots \cap fx_{n,C_n}$  (Def. 17 (42) and Def. 18 (43)). And since  $k_1, k_2 \in K_1$ , it follows that  $k_1, k_2 \in fx_{1,C_1} \cap fx_{2,C_2} \cap \dots \cap fx_{n,C_n}$ , and therefore that for all  $i \leq n$ ,  $k_1, k_2 \in fx_{i,C_i}$ . By the definition of equivalence classes associated with individual constraints (Def. 4 (13)), it then follows that for all  $i \leq n$ ,  $C_i(k_1) = C_i(k_2)$ . And then by the definition of the relation  $\approx_{\text{Com}}$  (Def. 25 (60)) we have  $k_1 \approx_{\text{Com}} k_2$ .



(ii) We have by assumption that  $k_1 \in K_1 \in K/C_{\text{Com}}$ ,  $k_3 \in K$ , and  $k_1 \approx_{\text{Com}} k_3$ . We need to show that then also  $k_3 \in K_1$ .

Consider first the assumption that  $k_1 \in K_1 \in K/C_{\text{Com}}$ . The existence of  $K_1$  implies that there is some  $n$ -tuple  $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle \in K/C_{\times}$  such that  $K_1 = \text{Intersect}(\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle) = fx_{1,C1} \cap fx_{2,C2} \cap \dots \cap fx_{n,Cn}$  (Def. 17 (42) and Def. 18 (43)). And since by assumption  $k_1 \in K_1$ , it follows that for all  $i \leq n$ ,  $k_1 \in fx_{i,Ci}$ , where  $fx_{i,Ci}$  is the equivalence class of candidate  $x$  in terms of constraint  $C_i$  (Def. 4 (13)).

Now consider the assumption that  $k_1 \approx_{\text{Com}} k_3$ . From the definition of  $\approx_{\text{Com}}$  (Def. 25 (60)), we have that for all  $i \leq n$ ,  $C_i(k_1) = C_i(k_3)$ . From the definition of the relation  $\approx_C$  (Def. 2 (9)) it then follows that for all  $i \leq n$ ,  $k_1 \approx_{C_i} k_3$ . And since  $\approx_{C_i}$  is an equivalence relation and therefore symmetric (Theorem 2 (12)), we also have  $k_3 \approx_{C_i} k_1$ . From the definition for equivalence classes associated with individual constraints (Def. 4 (13)), it then follows that for all  $i \leq n$ ,  $k_1, k_3 \in fx_{i,Ci}$ .

The equivalence classes of each constraint are collected into a quotient set (Def. 5 (15)). Therefore for all  $i \leq n$ , we have that  $fx_{i,Ci} \in K/C_{Ci}$ . From Theorem 6 (31) we know that these quotient sets are partitions on  $K$ , and therefore that the members of the quotient sets are disjoint. The equivalence class that  $k_1$  and  $k_3$  belong to for each constraint, is therefore also the only equivalence class that each of them belongs to for that constraint. This means that the Cartesian product taken over the quotient sets of the different constraints,  $K/C_{\times}$ , will contain one and only one  $n$ -tuple  $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle$  such that for all  $i \leq n$ ,  $k_1, k_3 \in fx_{i,Ci}$ .

We have seen above that  $k_1 \in K_1 \in K/C_{\text{Com}}$  implies the existence of some  $n$ -tuple  $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle \in K/C_{\times}$  such that for all  $i \leq n$ ,  $k_1 \in fx_{i,Ci}$ . But since only one such an  $n$ -tuple exists for  $k_1$ , it is also the  $n$ -tuple for which it is true that for all  $i \leq n$ ,  $k_3 \in fx_{i,Ci}$ . Then we have that for all  $i \leq n$ ,  $k_3 \in fx_{1,C1} \cap fx_{2,C2} \cap \dots \cap fx_{n,Cn}$ . And by the definition of *Intersect* (Def. 17 (42)) it follows that  $k_3 \in \text{Intersect}(\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle)$ . But as shown above,  $K_1 = \text{Intersect}(\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle)$ . Therefore  $k_3 \in K_1$ .

Then we finally have that  $k_1 \in K_1 \in K/C_{\text{Com}}$  and  $k_1 \approx_{\text{Com}} k_3$ , implies  $k_3 \in K_1$ .  $\square$

It is therefore possible to fully define the members of the set  $K/C_{\text{Com}}$  in terms of the equivalence relation  $\approx_{\text{Com}}$ .<sup>24</sup> This means that the candidates that are members of any given one of the sets in  $K/C_{\text{Com}}$  all have exactly the same number of violations in terms of every constraint. As far as the grammar is concerned, these candidates are all exactly the same. If our purpose is to describe the grammatical competence of the language user, this means that we can treat all candidates in each of the sets in  $K/C_{\text{Com}}$  alike, as if they are actually one single candidate – since according the grammar they are identical.

We can again check whether Theorem 12 is true of the example that we have been discussing throughout this chapter. The set  $K/C_{\text{Com}}$  in this example is  $K/C_{\text{Com}} = \{\{c_3\}, \{c_1, c_4\}, \{c_5\}, \{c_2\}\}$  (see (51) above). There is only one member of this set that contains more than one candidate, namely  $\{c_1, c_4\}$ . If Theorem 12 is true of this example, then the two candidates  $c_1$  and  $c_4$  have to receive the same number of violations in terms of every constraint. Referring back to the tableau in (6) confirms that this is indeed true. We have

---

<sup>24</sup> The set  $K/C_{\text{Com}}$  is therefore also the quotient set of the relation  $\approx_{\text{Com}}$  on the candidate set.

$C_1(c_1) = C_1(c_4) = 1$ ,  $C_2(c_1) = C_2(c_4) = 2$ , and  $C_3(c_1) = C_3(c_4) = 0$ . In this example the grammar can therefore not distinguish between these two candidates.

#### **4. Summary**

This chapter has investigated the formal properties of EVAL, the evaluative component of an OT grammar. The most important result of this chapter is that it established that a rank-ordering model of EVAL is entirely consistent with the basic architecture of a classic OT grammar. No changes or additions were made to the classic OT architecture in the development of this model. The rank-ordering model simply utilizes information that a classic OT grammar anyway generates. This chapter has developed a rigorous mathematical model of how the information generated by a classic OT grammar can be processed in order to establish a harmonic rank-ordering on the complete candidate set.

More specifically, this chapter has shown that: (i) The full candidate set is rank-ordered with respect to every constraint, and that this ordering is a chain with a minimum. (ii) The orderings of individual constraints can be combined to result in a final conglomerate ordering for the whole grammar. I showed that this conglomerate ordering agrees with our intuitions about the workings of an OT grammar (it abides by the principle of “strictness of strict domination”). The characteristics of this conglomerate ordering were investigated, and it was shown that it is an ordering on the full candidate set, that it is a chain ordering, and that it is guaranteed to have a minimum.

These results are important for our understanding of an OT grammar. Below I mention some of the most important perspectives on an OT grammar that this model

provides, and where appropriate I indicate how I use that result in the rest of the dissertation.

(i) *Richness of information.* The model developed in this chapter shows that a classic OT grammar generates a wealth of information about the relationships between the candidates. The information generated goes much further than simply distinguishing between the single optimal candidate and the losers. The grammar actually generates a much more detailed data structure that includes information about the harmonic relationship between any two candidates – i.e. also between candidates that are usually considered as losers.

An important question that this raises is whether this information is grammatically relevant. If the answer to this question is in the negative, then a classic OT grammar is much too powerful. It then massively over generates information. A more economic grammar would then have been one that would not have generated this information at all.

However, this chapter has shown that this information follows from the basic primitives of a classic OT grammar (constraint violation, constraint ranking, harmonic comparison between candidates). To formulate an alternative version of an OT grammar that does not generate this information will therefore be very difficult, if not impossible. This leads to the conclusion that it might be better to go the other direction – to assume that this information generated by the grammar is grammatically relevant.

This is the route that I take in this dissertation. I show several spheres in language performance where language users access and use this information. In variable phenomena language users access more candidates than just the single best candidate. They also access some of the candidates that are traditionally considered to be losers.

However, this implies that the “losers” cannot be a large amorphous group – then there would be no information on which language users can rely to select from among the “losers”. The information about the relationships between the losers is therefore crucial.

Also when rating the well-formedness of non-words language users access and use the richer information structure imposed by EVAL on the candidate set. Non-words that language users rate as more well-formed are simply non-words that correspond to candidates that occupy a higher slot in the rank-ordering that EVAL imposes on the candidate set. Similarly, I show that language users also use this information in lexical decision tasks. The lower a non-word occurs in the rank-ordering that EVAL imposes on the candidate set, the less seriously language users will consider the possibility that it is a word. Non-words corresponding to candidates lower in the rank-ordered candidate set, are therefore rejected quicker than non-words corresponding to candidates higher in the rank-ordered candidate set.

(ii) *No indeterminacy*. EVAL imposes two kinds of orderings on the candidate set – first EVAL orders the candidate set with regard to individual constraints (§2), and secondly EVAL orders the candidate set with regard to the whole grammar (§3). Both of these orderings have been shown to be on the whole candidate set (Theorem 6 (31) and Theorem 10 (59)). It has also been established that both of these orderings are chains (Theorem 4 (27) and Theorem 8 (57)). These two facts together imply that there is never indeterminacy in an OT grammar. It is always possible to determine for any two candidates how they are related to each other in terms of their harmony, whether with respect to an individual constraint or with respect to the whole grammar.

(iii) *A guaranteed output/unique access point.* The chain that EVAL imposes on the candidate set is guaranteed to have a minimum – again this holds of both the ordering with respect to individual constraints (Theorem 5 (29)) and with respect to the grammar as a whole (Theorem 9 (58)). Especially the fact that the chain associated with the whole grammar is guaranteed to have a minimum is relevant. This means that an OT grammar will always select a best candidate (or best set of candidates) from the candidate set. OT is therefore not a theory in which a derivation can crash because some input cannot be mapped onto any output candidate. OT is a forced-choice theory of grammar – the grammar is forced to select from the candidate set a best candidate.

There is another reason why it is important that the chain ordering imposed on the candidate set is guaranteed to have a minimum. This ensures that there is a uniquely identifiable point in terms of which access to the rank-ordered candidate set can be defined. A chain ordering with neither a maximum nor a minimum is both infinitely ascending and infinitely descending. If the chain ordering imposed by EVAL on the candidate set had neither a maximum nor a minimum, then no candidate could be uniquely identified with respect to its position in the chain ordering. For any candidate there would always be infinitely many candidates above it and infinitely many candidates below it. However, since the rank-ordered candidate set is guaranteed to have a minimum, there is a point on the candidate set that can be uniquely identified simply with respect to its position in the rank-ordering. In the rest of the dissertation I argue that the language user accesses the rank-ordered candidate set from its minimum. The minimum in the chain-ordering (the best candidate) is the most accessible, and the accessibility of candidates decreases the lower down they occur in the chain-ordering. In a variable

phenomenon the candidate that occupies the minimum position is therefore the most frequently observed output, the candidate that occupies the second slot is the second most frequently observed output, etc. In gradient well-formedness judgments, the candidate that occupies the minimum slot is rated best, and the further down from the minimum a candidate occurs, the less well-formed it is judged to be. In lexical decision tasks the candidate that occupies the minimum position in the chain-ordering is considered most seriously as a potential word, and is therefore associated with the slowest rejection times. The further down from the minimum a candidate occurs, the less seriously language users entertain the possibility that it might be a word, so that such candidates are associated with quicker rejection times.

(iv) *Independence from the candidate set.* The results of this chapter all assumed the existence of a set of candidates to be compared. However, the origin of this candidate set never featured in any of the proofs. The information structure that EVAL imposes on the candidate set therefore does not depend on the origin of the candidate set. EVAL can compare and order any set of candidate forms.

This result is also employed in the rest of the dissertation. We usually think of an OT grammar as comparing different output candidates for a single input – i.e. an input goes into GEN, and GEN then generates a set of candidate outputs for this input. However, the way in which EVAL works does not require the candidate set to originate in this manner. In particular, I argue that EVAL can compare candidates that are not morphologically related – that do not share the same input. GEN then generates a candidate set for several morphologically unrelated inputs, and we select subsets from these generated candidate sets to form a new set of candidates that is submitted to EVAL

for comparison (see Chapter 1 §1.2). A variable process does not always apply at the same rate in different contexts. In order to account for this, comparison across contexts is necessary, and I argue that this is achieved by allowing EVAL to compare candidates from different inputs (from different contexts).

Also when non-words are compared, whether in a well-formedness judgment task or in a lexical decision task, the candidates that are compared are not candidates generated by GEN for some input. In accounting for these kinds of phenomena I also rely on the fact that EVAL can compare any set of candidate forms.



## Appendix A: Definitions

### (3) **Constraints as relations between the candidate set and $\mathbb{U}$**

Let CON be the universal set of constraints, and  $K$  the set of candidates to be evaluated. Then,  $\forall C \in \text{CON}$ :

$C: K \rightarrow \mathbb{U}$  such that  $\forall k \in K, C(k) =$  number of violations of  $k$  in terms of  $C$

### (4) **Def. 1: Functions**

A relation  $R$  from  $A$  to  $B$  is a function iff:

- (a) the domain of  $R = A$  (i.e. every member of  $A$  is mapped onto some member of  $B$ ), and
- (b) each element in  $A$  is mapped onto just one element in  $B$  ( $R$  is single valued).

### (9) **Def. 2: The relation $\approx_C$ on $K$**

Let  $K$  be the candidate set to be evaluated by EVAL, and CON the set of constraints.

Then, for all  $k_1, k_2 \in K$ , and for all  $C \in \text{CON}$ , let:

$$k_1 \approx_C k_2 \quad \text{iff} \quad C(k_1) = C(k_2).$$

### (11) **Def. 3: An equivalence relation**

A binary relation  $R$  on some set is an equivalence relation on that set iff  $R$  is

(i) reflexive, (ii) symmetric, and (iii) transitive.

### (13) **Def. 4: Equivalence classes on $K$ in terms of $\approx_C$**

For all  $k_1 \in K, fk_{1,C} := \{k_2 \in K \mid k_1 \approx_C k_2\}$

(15) **Def. 5: Quotient set on  $K$  modulo  $\approx_C$**

$$K/C := \{fk_{\cdot,C} \mid k \in K\}$$

(17) **Def. 6: The ordering relation  $\leq_C$  on the set  $K/C$**

For all  $C \in \text{CON}$  and all  $fk_{1,C}, fk_{2,C} \in K/C$ :

$$fk_{1,C} \leq_C fk_{2,C} \text{ iff } C(k_1) \leq C(k_2).$$

(19) **Def. 7: The ordering relation  $\leq_{C'}$  on the set  $K$**

For all  $C \in \text{CON}$  and all  $k_1, k_2 \in K$ :

$$k_1 \leq_{C'} k_2 \text{ iff } C(k_1) \leq C(k_2)$$

(21) **Def. 8: An order-embedding**

Let  $P$  and  $Q$  be ordered sets. A map  $\varphi: P \rightarrow Q$  is said to be an order-embedding if

$x \leq y$  in  $P$  iff  $\varphi(x) \leq \varphi(y)$  in  $Q$ .

(22) **Def. 9: A mapping from  $\langle K/C, \leq_C \rangle$  to  $\langle K, \leq_{C'} \rangle$**

$\psi: \langle K/C, \leq_C \rangle \rightarrow \langle K, \leq_{C'} \rangle$  such that:

$$\text{For all } fk_{x,C} \in K/C \text{ and for all } k_y \in fk_{x,C}, \psi(fk_{x,C}) = k_y.$$

(26) **Def. 10: Definition of a chain**

Let  $P$  be an ordered set. Then  $P$  is a chain iff for all  $x, y \in P$ , either  $x \leq y$  or  $y \leq x$ .

(28) **Def. 11: Minimum of an ordered set**

Let  $P$  be an ordered set and  $Q \subseteq P$ . Then:

$a \in Q$  is the minimum of  $Q$  iff  $a \leq x$  for every  $x \in Q$ .

(30) **Def. 12: A partition**

A set  $P$  is said to be a partition on some set  $A$  iff:

- (a)  $P$  consists of non-empty subsets of  $A$ .
- (b) The sets in  $P$  are exhaustive – each element of  $A$  is in some set in  $P$ .
- (c) The sets in  $P$  are disjoint – no two different sets in  $P$  have any element in common.

(34) **Def. 13: Cartesian product**

Let  $I$  be the set  $\{1, 2, \dots, n\}$ , the *index set*, and let  $H$  be a function with domain  $I$ . Then, for each  $i \in I$ , we have a set  $H(i)$ . The Cartesian product of  $H(i)$  for all  $i \in I$  is defined as follows:

$$\mathbf{V}_{i \in I} H(i) := \{f \mid f \text{ is a function with domain } I \text{ and } \forall i (i \in I \rightarrow f(i) \in H(i))\}$$

(35) **Def. 14: Step 1 in combination process = Cartesian product between sets  $K/C_i$**

Let  $I$  be the set  $\{1, 2, \dots, n\}$ , the *index set*, such that  $\|C_1 \circ C_2 \circ \dots \circ C_n\|$ .

Let  $K/C_i$  be the quotient on  $K$  associated with  $C_i$ . We want the Cartesian product of all the quotient sets. We define this as follows:

$$\mathbf{V}_{i \in I} K/C_i := \{f \mid f \text{ is a function with domain } I \text{ and } \forall i (i \in I \rightarrow f(i) \in K/C_i)\}$$

The set  $\mathbf{V}_{i \in I} K/C_i$  will be referred to as  $K/C_{\times}$ .

(37) **Def. 15: Lexicographic order**

Let  $\mathbf{V}_{i \in I} H(i)$  be the set as defined in Def. 13 (34) above, and let  $\langle x_1, x_2, \dots, x_n \rangle, \langle y_1, y_2, \dots, y_n \rangle \in \mathbf{V}_{i \in I} H(i)$ .

The lexicographic order on  $\mathbf{V}_{i \in I} H(i)$  is defined as follows:

((37) continued)

$\langle x_1, x_2, \dots, x_n \rangle \leq \langle y_1, y_2, \dots, y_n \rangle$  iff:

(i) For all  $i \leq n$ ,  $x_i = y_i$  (then  $\langle x_1, x_2, \dots, x_n \rangle = \langle y_1, y_2, \dots, y_n \rangle$ )

OR (ii)  $\exists k$  such that:

- $\forall i (i < k \rightarrow x_i = y_i)$ , and
- $x_k < y_k$  (then  $\langle x_1, x_2, \dots, x_n \rangle < \langle y_1, y_2, \dots, y_n \rangle$ )

(38) **Def. 16: Step 2 in the combination process = ordering  $K/C_x$**

Let  $C_i \in \text{CON}$ , with the ranking  $\|C_1 \circ C_2 \circ \dots \circ C_n\|$ , and  $K/C_i$  the quotient set associated with constraint  $C_i$  (as defined in Def. 5 (15)). Let  $f_{x_i, C_i}, f_{y_i, C_i} \in K/C_i$  be the equivalence classes of candidates  $x_i$  and  $y_i$  in terms of constraint  $C_i$  (as defined in Def. 4 (13)).

Let  $\leq_{C_i}$  be the ordering that EVAL imposes on the candidate set in terms of constraint  $C_i$  (as defined in Def. 6 (17)).

Let  $K/C_x$  be the Cartesian product of  $K/C_i$  for all  $i \in I$  (as defined in Def. 14 (35)).

Let  $\langle f_{x_1, C_1}, f_{x_2, C_2} \dots f_{x_n, C_n} \rangle, \langle f_{y_1, C_1}, f_{y_2, C_2} \dots f_{y_n, C_n} \rangle \in K/C_x$ .

Then  $\leq_x$ , the lexicographic order on  $K/C_x$ , is defined as follows:

$\langle f_{x_1, C_1}, f_{x_2, C_2} \dots f_{x_n, C_n} \rangle \leq_x \langle f_{y_1, C_1}, f_{y_2, C_2} \dots f_{y_n, C_n} \rangle$  iff:

(i)  $\forall i (i \leq n \rightarrow f_{x_i, C_i} =_{C_i} f_{y_i, C_i})$

(then  $\langle f_{x_1, C_1}, f_{x_2, C_2} \dots f_{x_n, C_n} \rangle =_x \langle f_{y_1, C_1}, f_{y_2, C_2} \dots f_{y_n, C_n} \rangle$ )

OR (ii)  $\exists k$  such that:

- $\forall i (i < k \rightarrow f_{x_i, C_i} =_{C_i} f_{y_i, C_i})$ , and

- $f_{x_k, C_k} <_{C_k} f_{y_k, C_k}$ .

(then  $\langle f_{x_1, C_1}, f_{x_2, C_2} \dots f_{x_n, C_n} \rangle <_x \langle f_{y_1, C_1}, f_{y_2, C_2} \dots f_{y_n, C_n} \rangle$ )

(42) **Def. 17:** First half of step 3 in the combination process = *Intersect*

Let  $K/C_x$  be the set as defined in Def. 14 (35) above, and let  $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle \in K/C_x$ . Then we define *Intersect*:  $K/C_x \rightarrow \wp(K)$  as follows:

*Intersect*( $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle$ ) is undefined if  $fx_{1,C1} \cap fx_{2,C2} \cap \dots \cap fx_{n,Cn} = \emptyset$ , and *Intersect*( $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle$ ) =  $fx_{1,C1} \cap fx_{2,C2} \cap \dots \cap fx_{n,Cn}$  otherwise.

(43) **Def. 18:** Collecting the output of *Intersect* into one set

$K/C_{Com} := \{ Z \mid \exists \langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle \in K/C_x, \text{ such that}$

$$Z = \text{Intersect}(\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle) \}$$

(45) **Def. 19:** Second half of Step 3 in the combination process:

the ordering  $\leq_{Com}$  on  $K/C_{Com}$ .

Let  $(fx_{1,C1} \cap fx_{2,C2} \cap \dots \cap fx_{n,Cn}), (fy_{1,C1} \cap fy_{2,C2} \cap \dots \cap fy_{n,Cn}) \in K/C_{Com}$ .

Then  $\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle, \langle fy_{1,C1}, fy_{2,C2} \dots fy_{n,Cn} \rangle \in K/C_x$ .

Then we define the order  $\leq_{Com}$  on  $K/C_{Com}$  as follows:

$$(fx_{1,C1} \cap fx_{2,C2} \cap \dots \cap fx_{n,Cn}) \leq_{Com} (fy_{1,C1} \cap fy_{2,C2} \cap \dots \cap fy_{n,Cn}) \text{ iff}$$

$$\langle fx_{1,C1}, fx_{2,C2} \dots fx_{n,Cn} \rangle \leq_x \langle fy_{1,C1}, fy_{2,C2} \dots fy_{n,Cn} \rangle.$$

(47) **Def. 20:** Inverse

Let  $A$  and  $B$  be two sets, and  $F: A \rightarrow B$  a relation on  $A \times B$ .  $F$  can then be represented as set of ordered pairs,  $F = \{ \langle a_1, b_1 \rangle, \langle a_2, b_2 \rangle, \dots \}$  with  $a_i \in A$  and  $b_i \in B$ .

$F^{-1}: B \rightarrow A$ , the inverse of  $F$ , is then a relation on  $B \times A$ , and is defined as follows:

$$F^{-1} := \{ \langle b, a \rangle \mid \langle a, b \rangle \in F \}$$

(48) **Def. 21: The inverse of *Intersect***

Let  $K' \subseteq K$ . Then we can define  $Intersect^{-1}$  as follows:

$Intersect^{-1}: \wp(K) \rightarrow K/C_{\times}: Intersect^{-1}(K') = \langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle$  iff  
 $Intersect(\langle fx_{1,C_1}, fx_{2,C_2} \dots fx_{n,C_n} \rangle) = K'$ .

(49) **Def. 22: An order preserving mapping**

Let  $P$  and  $Q$  be ordered sets. A map  $\varphi: P \rightarrow Q$  is said to order preserving if  $x \leq y$  in  $P$  implies  $\varphi(x) \leq \varphi(y)$  in  $Q$ .

(53) **Def. 23: Crucial constraints**

Let  $k_1, k_2 \in K$ , and let the grammar under consideration be  $\|C_1 \circ C_2 \circ \dots \circ C_n\|$ .

Then we define  $Crux_{1,2}$ , the crucial constraint for  $k_1$  and  $k_2$ , as follows:

$Crux_{1,2} = C_i$  such that  $C_i(k_1) \neq C_i(k_2)$  and  $\neg \exists j (j < i \text{ and } C_j(k_1) \neq C_j(k_2))$ .

(54) **Def. 24: Strictness of domination with reference to  $\langle K/C_{Com}, \leq_{Com} \rangle$**

Let  $k_1, k_2 \in K$ , and  $K_1, K_2 \in K/C_{Com}$  such that  $k_1 \in K_1$  and  $k_2 \in K_2$ .

Let  $fk_{i,C_j}$  be the equivalence class of  $k_i \in K$  in terms of constraint  $C_j$  as defined in

Def. 4 (13) above, and  $\leq_{C_j}$  the ordering associated with this constraint as defined in Def. 6 (17) above.

Let  $Crux_{1,2}$  be the crucial constraint as defined just above in Def. 23 (53), and

$\leq_{Crux_{1,2}}$  the ordering that EVAL imposes on the candidate set with reference to  $Crux_{1,2}$ .

Then:

$$K_1 <_{Com} K_2 \text{ iff } fk_{1,Crux_{1,2}} <_{Crux_{1,2}} fk_{2,Crux_{1,2}}.$$

(60) **Def. 25:  $\approx_{\text{Com}}$  as a relation on  $K$**

Let the grammar under consideration be  $\|C_1 \circ C_2 \circ \dots \circ C_n\|$ , and  $k_1, k_2 \in K$ .

Then we define  $\approx_{\text{Com}}$  as follows:

$$\approx_{\text{Com}} \subseteq K \times K \text{ such that } k_1 \approx_{\text{Com}} k_2 \text{ iff for all } i \leq n \ C_i(k_1) = C_i(k_2)$$

**Appendix B: Theorems and Lemmas**

(5) **Theorem 1: Constraints as functions**

All constraints are functions.

(12) **Theorem 2:  $\approx_C$  as an equivalence relation**

For all  $C \in \text{CON}$ ,  $\approx_C$  is an equivalence relation on  $K$ .

(24) **Theorem 3: That  $\psi$  is an order-embedding**

The mapping  $\psi$  as defined in Def. 9 (16) is an order-embedding.

(27) **Theorem 4: That  $\leq_C$  defines a chain**

The ordering that  $\leq_C$  imposes on  $K/C$  is a chain.

(29) **Theorem 5: That  $\langle K/C, \leq_C \rangle$  has a minimum**

The ordering  $\leq_C$  always has a minimum in  $K/C$ .

(31) **Theorem 6:  $K/C$  as a partition on  $K$**

$K/C$  is a partition on  $K$ .

- (40) **Lemma 1: That  $\leq_x$  defines a chain**  
 $\leq_x$  defines a chain on  $K/C_x$ .
- (41) **Lemma 2: That  $\leq_x$  always has a minimum**  
The ordering  $\leq_x$  always has a minimum in  $K/C_x$ .
- (50) **Lemma 3: That  $Intersect^{-1}$  is order preserving**  
 $Intersect^{-1}$  is an order preserving mapping.
- (55) **Theorem 7: Strictness of domination and  $\langle K/C_{Com}, \leq_{Com} \rangle$**   
 $\langle K/C_{Com}, \leq_{Com} \rangle$  abides by strictness of domination.
- (57) **Theorem 8: That  $\leq_{Com}$  defines a chain**  
The ordering that  $\leq_{Com}$  imposes on  $K/C_{Com}$  is a chain.
- (58) **Theorem 9: That  $\langle K/C_{Com}, \leq_{Com} \rangle$  always has a minimum**  
The ordering  $\leq_{Com}$  always has a minimum in  $K/C_{Com}$ .
- (59) **Theorem 10: That  $K/C_{Com}$  is a partition on  $K$**   
 $K/C_{Com}$  is a partition on  $K$ .
- (61) **Theorem 11: That  $\approx_{Com}$  is an equivalence relation**  
 $\approx_{Com}$  is an equivalence relation on  $K$ .



(62) **Theorem 12:** The members of  $K/C_{\text{Com}}$  can be fully defined in terms of  $\approx_{\text{Com}}$

Let  $K$  be the candidate set,  $K_1 \in K/C_{\text{Com}}$  and  $k_1 \in K_1$ . Then:

- (i)  $\forall k_2 \in K_1, k_1 \approx_{\text{Com}} k_2$ , and
- (ii)  $\forall k_3 \in K$  such that  $k_1 \approx_{\text{Com}} k_3, k_3 \in K_1$ .