

Gradient constraints in finite state OT: The unidirectional and the bidirectional case

Gerhard Jäger

1. Introduction^{*}

Optimality Theory (Prince & Smolensky 1993) has offered a novel unifying perspective on different branches of linguistic research. It was originally conceived as a form of synchronic grammatical description in the tradition of generative grammar, and the first domain of application was phonology. In the meantime optimality theoretic concepts have been used for all modules of grammar, ranging from phonology over morphology and syntax to semantics and pragmatics. The idea that a grammar is basically a ranking of a small set of universal soft constraints lends itself easily to an explanatory approach to typological variation. Furthermore, Optimality Theory (OT henceforth) has been used as a guideline in recent research on learnability (Tesar & Smolensky 2001, Boersma & Hayes 2001), language acquisition (Hayes 1999), and diachronic change (Haspelmath 1999).

In its standard form, OT is an abstract model of language production. The grammar generates a (possibly large) set of potential surface realizations of a given underlying form, and the evaluation algorithm picks out the optimal candidate(s) from this set. Several researchers have argued though that the notion of optimization also plays a role in language comprehension, and that expressive and interpretative optimization actually work in tandem. Bidirectional versions of OT have been considered for synchronic description of morphological (Wunderlich 2001a,b) and syntactic phenomena (by Wilson 2001 for binding and by Lee 1999 for word order freezing effects) as well as in the realm of semantics and pragmatics (see for instance Egg 1999, Jäger & Blutner 2000, von Stechow 2000 on German adverbials, Beaver 2000 on anaphora resolution, and Zeevat 1999 on presuppositions). The recent work of Reinhard Blutner (especially Blutner 2001) represents the presumably best worked out version of Bidirectional OT.

* I would like to thank Dieter Wunderlich for an invitation to Düsseldorf that gave me the opportunity to discuss the material from this paper as well as the issue of bidirectional OT in general. Especially the discussions with Dieter himself, with Wiebke Petersen and with Barbara Stiebels were very inspiring. Furthermore I am grateful to the editors of this volume and an anonymous referee for valuable comments on a previous version of this paper.

One might argue that theories like OT that assume a competition between potentially large different linguistic entities to determine the grammaticality of one single linguistic object are computationally too costly to be psychologically realistic models of linguistic competence. This criticism is certainly valid. OT as such places no upper bound on the computational complexity of natural languages. It is an easy exercise to design OT grammars that are undecidable. This is in stark contrast with the fact that the generative complexity of natural languages is known to be quite limited. According to the available evidence, syntax does not go beyond the class of mildly context sensitive languages (see Joshi et al. 1990), while most of phonology and morphology can even be described by means of finite state techniques (as pointed out by Kaplan & Kay 1981). Ideally, a plausible grammatical formalism should exceed the necessary generative capacity as little as possible. The success of the research paradigm of OT thus crucially depends on the issue whether it is possible to impose constraints on possible OT systems in such a way that the resulting grammars are computationally tractable (and thus psychologically plausible).

The computational complexity of OT is currently an area of active research, and several results have been obtained in the literature. Most importantly, Frank & Satta (1998) show that under certain general restrictions, (unidirectional) optimization is a finite state technique. This means that an OT system can be implemented as a finite state transducer provided the generator can be modeled by means of finite state techniques and all constraints are regular languages. In other words, if all components of an OT-system are finite state objects, the system as a whole is so too. In Jäger (2001) it is shown that this result directly carries over to bidirectional OT.

These results are confined to OT systems that only use binary constraints. Many empirical works in OT assume gradient constraints though, i.e. constraints that can be violated arbitrarily many times. Frank & Satta (1998) also prove that the relation of optimal input-output pairings may exceed the power of finite state techniques even if all components of the OT system in question are finite state implementable if gradient constraints are involved. This negative result is countered by a positive result in Gerdemann and van Noord (2000). These authors present a finite state implementation of a certain OT system involving gradient constraints.

The present paper contains two main results. After reviewing the work of Frank & Satta and of Gerdemann & van Noord on the complexity of unidirectional OT as well as the results from Jäger (2001) on bidirectional OT, I will propose a generalization of Gerdemann & van Noord's construction for the unidirectional case, i.e. I will give general sufficient (though not necessary) conditions when an OT system using gradient constraints can be implemented as a finite state automaton. Furthermore, it will be shown that this result does not carry over to the bidirectional case. An OT system will be presented that remains within the limits of finite state techniques if unidirectional optimization is used but which exceeds this complexity class if bidirectional optimization is applied. The latter result thus shows that bidirectional OT is intrinsically more complex than unidirectional OT.

2. Basic concepts of finite state OT

Let me first make precise what is meant by the term “OT-systems.” In the general case, an OT-system consists of a relation GEN and a finite set of constraints that are linearly ordered by some constraint ranking.¹ Constraints may be violated several times. So a constraint should be construed as a function from GEN into the natural numbers. Thus an OT-system assigns every pair in GEN a finite sequence of natural numbers. The ordering of the elements of GEN that is induced by the OT-system is according to the lexicographic ordering of these sequences. This leads to the following definition:

Definition 1 (OT-System)

1. An OT-system is a pair $\mathcal{O} = \langle \text{GEN}, C \rangle$, where GEN is a relation, and $C = \langle c_1, \dots, c_p \rangle$, $p \in \mathbb{N}$ is a linearly ordered sequence of functions from GEN to \mathbb{N} .
2. Let $a, b \in \text{GEN}$. $a <_{\mathcal{O}} b$ iff there is an i with $1 \leq i \leq p$ such that $c_i(a) < c_i(b)$ and for all $j < i$: $c_j(a) = c_j(b)$.

The following lemma states a property of the ranking of candidates induced by an OT-system that will be important later on. We call a binary relation R “weakly linear” iff R is well-founded, and for all x, y : xRy or yRx or $x \equiv_R y$, where $x \equiv_R y$ iff for all z : zRx iff zRy .

Lemma 1 Let \mathcal{O} be an OT-system. Then $<_{\mathcal{O}}$ is weakly linear.

Proof: We assign every element of GEN an ordinal number by the function f that is defined by

$$f(x) = \sum_{i=1}^p w^{p-i} \cdot c_i(x)$$

It is easy to see that $x <_{\mathcal{O}} y$ iff $f(x) < f(y)$. Since the ordering of the ordinal numbers is well-founded, so is $<_{\mathcal{O}}$. Furthermore, $x \equiv_{<_{\mathcal{O}}} y$ iff $f(x) = f(y)$. So if $x \not<_{\mathcal{O}} y$ and $y \not<_{\mathcal{O}} x$, it must be the case that $f(x) = f(y)$, hence $x \equiv y$. \dashv

Intuitively, an output o is an (unidirectionally) optimal output for some input i iff GEN relates i and o , and o is optimal among the possible outputs for i . This is expressed by the following definition:

Definition 2 (Unidirectional optimality) Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ be an OT-system. Then $\langle i, o \rangle$ is unidirectionally optimal with respect to \mathcal{O} iff

1. $\langle i, o \rangle \in \text{GEN}$
2. there is no o' such that $\langle i, o' \rangle \in \text{GEN}$ and $\langle i, o' \rangle <_{\mathcal{O}} \langle i, o \rangle$

1. Some authors only require the constraints to be partially ordered. Since a given candidate is optimal according to some partial ordering iff it is optimal according to all total extensions of this partial ordering, the results obtained in this paper can easily be extended to this more general setup.

In most research papers on OT, the candidate sets that are taken under consideration are finite and even fairly small, and the search for the optimal candidate is done manually by comparing the patterns of constraint violations. It has frequently been observed that in realistic applications, candidate sets might be very large, which would render this kind of naive brute force algorithm computationally very expensive. Even worse, if the candidate set is infinite, there is no guarantee that this kind of algorithm terminates. Thus the success of the OT research program crucially hinges on the issue whether there are computationally tractable evaluation algorithms.

It is obvious that the complexity of the task of finding the optimal candidates for a given OT-system depends on the complexities of the generator and of the constraints. In the general case, these will provide a lower bound for the complexity of the OT-system as a whole, both in terms of automata theoretic complexity and in terms of resource complexity. The crucial question is whether an OT-system as a whole may have a higher complexity than the most complex of its components. Furthermore, this issue may depend on the mode of evaluation that is chosen. For instance, unidirectional OT might be less complex than bidirectional OT.

While these issues are still open in the general case, the literature contains some promising results about the complexity of OT in cases where all components of the OT-system are finite state objects. These insights are of great practical importance in phonology and morphology, where finite state techniques are usually sufficiently expressive.

In syntax and semantics, this kind of result cannot be employed immediately since it is well-known that more automata-theoretic power is needed here. Nevertheless the finite state case is interesting since it indicates that the OT mechanism as such is not all that powerful after all.

In this section I will briefly review some basic properties of finite state objects, and I will discuss the most impressive piece of work on the complexity of OT, Frank & Satta's (1998) construction. This will pave the ground for the extrapolation of Frank & Satta's unidirectional case that is to be presented later on. In the subsequent discussion of finite state automata, finite state transducers, regular languages and rational relations, I will make heavy use of Roche & Schabes (1997). The interested reader is referred there for further information and references.

We assume that the reader is familiar with the basic concepts of a finite state automaton and a regular language and give the definition here for reference.

Definition 3 (FSA) A finite-state automaton A is a 5-tuple $\langle \Sigma, Q, i, F, E \rangle$, where Σ is a finite set called the *alphabet*, Q is a finite set of *states*, $i \in Q$ is the *initial state*, $F \subseteq Q$ is the set of final states, and $E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ is the set of *edges*.

Following standard practice, I use Σ^* to refer to the set of strings over the alphabet Σ , including the empty string. The letter ϵ symbolizes the empty string.

Definition 4 The *extended set of edges* $\hat{E} \subseteq Q \times \Sigma^* \times Q$ is the smallest set such that

1. $\forall q \in Q : \langle q, \mathbf{e}, q \rangle \in \hat{E}$
2. $\forall w \in \Sigma^*$ and $\forall a \in \Sigma \cup \{\mathbf{e}\}$: if $\langle q_1, w, q_2 \rangle \in \hat{E}$ and $\langle q_2, a, q_3 \rangle \in E$, then $\langle q_1, wa, q_3 \rangle \in \hat{E}$.

A finite-state automaton A defines the following language $L(A)$:

$$L(A) := \{w \in \Sigma^* \mid \exists q \in F : \langle i, w, q \rangle \in \hat{E}\}$$

If $\mathcal{L} = L(A)$, it is said that the FSA A *recognizes* the language \mathcal{L} . The class of *regular languages* is the class of languages that are recognized by some FSA.

A finite state transducer (FST) is a FSA that produces an output. Every edge of the automaton is labeled with an input and an output, where both input and output are strings over the input alphabet and the output alphabet respectively. An FST does not just recognize strings but transforms input strings in output strings.

Definition 5 (FST) A *Finite-State Transducer* is a tuple $\langle \Sigma_1, \Sigma_2, Q, i, F, E \rangle$ such that

- Σ_1 is a finite alphabet, namely the *input alphabet*.
- Σ_2 is a finite alphabet, namely the *output alphabet*.
- Q is a finite set of *states*.
- $i \in Q$ is the *initial state*.
- $F \subseteq Q$ is the set of *final states*.
- $E \subseteq Q \times \Sigma_1^* \times \Sigma_2^* \times Q$ is the set of *edges*.

The notion of an extended edge of a FST is analogous to the corresponding concept for FSA.

Definition 6 The *extended set of edges* $\hat{E} \subseteq Q \times \Sigma_1^* \times \Sigma_2^* \times Q$ is the smallest set such that

1. $\forall q \in Q : \langle q, \mathbf{e}, \mathbf{e}, q \rangle \in \hat{E}$
2. $\forall v_1, w_1 \in \Sigma_1^*$ and $\forall v_2, w_2 \in \Sigma_2^*$: if $\langle q_1, v_1, v_2, q_2 \rangle \in \hat{E}$ and $\langle q_2, w_1, w_2, q_3 \rangle \in E$, then $\langle q_1, v_1 w_1, v_2 w_2, q_3 \rangle \in \hat{E}$.

A finite-state transducer T defines the following relation between Σ_1^* and Σ_2^* :

$$R(A) = \{\langle v, w \rangle \in \Sigma_1^* \times \Sigma_2^* \mid \exists q \in F : \langle i, v, w, q \rangle \in \hat{E}\}$$

The class of relations that is defined by some FST is called the class of *rational relations*. A simple FST that implements the rational relation $\{\langle a^n, b^n c^* \rangle \mid n \in \mathbb{N}\}$ is given in figure 1 for illustration.

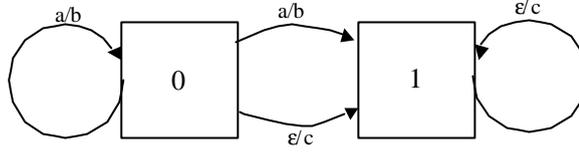


Figure 1: FST implementing the rational relation $\{\langle a^n, b^n c^* \rangle \mid n \in \mathbb{N}\}$

The classes of regular languages and of rational relations are subject to certain *closure properties*. $(R_1 \circ R_2)$ is the relation composition of R_1 and R_2 , i.e. $\{\langle v, w \rangle \mid \exists x (vR_1x \wedge xR_2w)\}$. R^\cup is the inverse of the relation R , i.e. $\{\langle w, v \rangle \mid vRw\}$. L_1L_2 is the concatenation of L_1 and L_2 , i.e. the set of strings $\{uv \mid u \in L_1 \wedge v \in L_2\}$. Likewise, R_1R_2 is the concatenation of the relations R_1 and R_2 , i.e. the relation $\{\langle u_1v_1, u_2v_2 \rangle \mid \langle u_1, u_2 \rangle \in R_1 \wedge \langle v_1, v_2 \rangle \in R_2\}$. \mathbf{I}_L is the identity relation on L , i.e. $\{\langle v, v \rangle \mid v \in L\}$. L^* , the Kleene closure of the language L , is the set of strings $\{w_1 \dots w_n \mid w_i \in L \text{ for all } 1 \leq i \leq n, n \geq 0\}$. Likewise, R^* , the Kleene closure of the relation R , is the relation $\{\langle u_1 \dots u_n, v_1 \dots v_n \rangle \mid \langle u_i, v_i \rangle \in R \text{ for all } 1 \leq i \leq n, n \geq 0\}$.

- Every finite language is regular.
- If L_1 and L_2 are regular languages, then $L_1 \cap L_2$, $L_1 \cup L_2$, $L_1 - L_2$, L_1L_2 , and L_1^* are also regular languages.
- If R_1 and R_2 are rational relations, then $R_1 \cup R_2$, $R_1 \circ R_2$, R_1^\cup , R_1R_2 , and R_1^* are also rational relations.
- If R is a rational relation, then $Dom(R)$ and $Rg(R)$ (the domain $\{x \mid \exists y : xRy\}$ and the range $\{y \mid \exists x : xRy\}$ of R) are regular languages.
- If L_1 and L_2 are regular languages, then $L_1 \times L_2$ and \mathbf{I}_{L_1} are rational relations.

Note that the rational relations are not closed under intersection and complement. Frank & Satta use these closure properties to show that for a significant class of OT-systems, unidirectional optimization is a rational relation provided all building blocks are rational.

3. Frank & Satta (1998)

Now let us turn our attention to Frank & Satta's construction. They restrict the class of OT-systems in two ways. First, OT constraints in general “count”, a given constraint may be violated arbitrarily many times. Frank & Satta restrict attention to binary constraints, i.e. constraints c with the property $Rg(c) = \{0, 1\}$. OT-systems which are not binary but have an upper limit for the number of constraint violations are implicitly covered; a constraint c that can be violated at most n times can be represented by n binary constraints of the form “Violate c less than i times” for $1 \leq i \leq n$. The ranking of these new constraints is inessential for the induced ordering relation.

Second, one may distinguish constraints that evaluate solely the output and constraints that properly evaluate an input-output pair. The former type of constraint is called *markedness constraints* in the literature (see for instance Kager 1999), while the latter are covered under the term *faithfulness constraint*. Let us make this precise. We use the term “Output Markedness Constraint” since markedness constraints may also evaluate solely the input. Such input constraints have no effect for unidirectional OT, but they become important later when bidirectionality will be discussed.

Definition 7 ((Output) Markedness Constraint) Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ be an OT-system. Constraint c_i is an *output markedness constraint* iff

$$\langle i, o \rangle \in \text{GEN} \wedge \langle i', o \rangle \in \text{GEN} \rightarrow c_i(\langle i, o \rangle) = c_i(\langle i', o \rangle)$$

Frank & Satta restrict attention to binary output markedness constraints. Obviously, these can be represented as languages over the output alphabet, namely as the set of outputs that obey them. The central part of their construction is an operation called *conditional intersection* that combines a relation with a language. Karttunen (1998) calls this operation *lenient composition*, and I will follow this terminology.

Definition 8 (Lenient composition) Let R be a relation and $L \subseteq Rg(R)$. The *lenient composition* $R \uparrow L$ of R with L is defined as

$$R \uparrow L \doteq (R \circ \mathbf{I}_L) \cup (\mathbf{I}_{\text{Dom}(R) - \text{Dom}(R \circ \mathbf{I}_L)} \circ R)$$

By applying the definitions, it is easy to see that $\langle x, y \rangle \in R \uparrow L$ iff xRy and either $y \in L$ or there is no $z \in L$ such that xRz . In other words, $\{y \mid \langle x, y \rangle \in R \uparrow L\}$ is the set of y s that are related to x by R , and that are optimal with respect to the constraint L . Furthermore, it follows from the closure properties given above that $R \uparrow L$ is a rational relation provided R is rational and L is a regular language.

Let us illustrate the working of lenient composition with an example. Let R be a relation $\{\langle a, x \rangle, \langle a, y \rangle, \langle b, z \rangle, \langle b, w \rangle, \langle c, u \rangle, \langle d, v \rangle\}$. Suppose there is a constraint c that is obeyed by y, z and u , while x, w and v violate it. This constraint can be represented by the language $L = \{y, z, u\}$. Figure 2 illustrates this scenario.

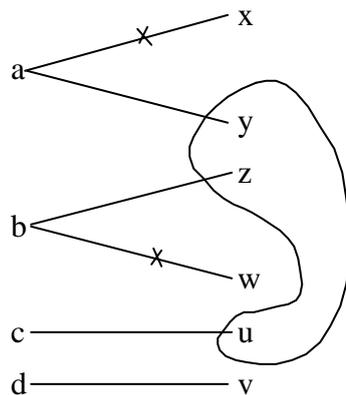


Figure 2

Which input-output pairs are optimal in this setup? For the input a , the only optimal output is y because it is the only output for a that obeys the constraint. Likewise, the only optimal outputs for b and c are z and u respectively. There is no output for d that obeys the constraint. Hence v is an optimal output for d .

Lenient composition gives exactly this result. $R \circ \mathbf{I}_L = \{\langle a, y \rangle, \langle b, z \rangle, \langle c, u \rangle\}$, i.e. the range of R is restricted to L . Next the domain of this relation is formed: $\text{Dom}(R \circ \mathbf{I}_L) = \{a, b, c\}$. This is the set of inputs that has a constraint obeying output. The complement of this set, the set of inputs that do not have a constraint obeying output, is given by $\text{Dom}(R) - \text{Dom}(R \circ \mathbf{I}_L) = \{d\}$. Restricting R to this input yields $\mathbf{I}_{\text{Dom}(R) - \text{Dom}(R \circ \mathbf{I}_L)} \circ R = \{\langle d, v \rangle\}$. The set of optimal input-output pairs is given by the union of this relation with the relation $R \circ \mathbf{I}_L$ (the set of pairs that obey the constraint):

$$R \uparrow L = \{\langle a, y \rangle, \langle b, z \rangle, \langle c, u \rangle, \langle d, v \rangle\}$$

Unidirectional optimality can now be implemented in a straightforward way, namely by successively leniently composing the (binary markedness) constraints of an OT-system with GEN.

Theorem 1 (Frank & Satta) Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ with $C = \langle c_1, \dots, c_p \rangle$ be an OT-system such that C solely consists of binary output markedness constraints. Then $\langle i, o \rangle$ is unidirectionally optimal iff $\langle i, o \rangle \in \text{GEN} \uparrow c_1 \dots \uparrow c_p$.

The proof of this theorem is obvious from the definitions. Crucially, it follows that unidirectional optimality is a rational relation provided GEN is rational and all constraints are regular languages.

4. Gradient constraints in unidirectional OT

Despite the theoretical importance of Frank & Satta's result, the restriction to binary constraints limits its practical use in quite a significant way. This shortcoming is somewhat alleviated by the observation (also due to Frank & Satta) that their result readily extends to gradient constraints which impose an upper limit on the number of violations. Nevertheless, in the general case gradient constraints extend the generative power of an OT-system beyond the limits of finite state techniques. Frank & Satta illustrate this fact by means of an example construction which is based on an idea by Markus Hiller. Consider the following simple OT-system which has just one constraint, c . (Here and throughout the rest of the paper, the discussion is confined to markedness constraints. Thus constraints are construed as functions that operate solely either on the input or on the output.)

$$\begin{aligned} \text{GEN} &= \{\langle a^n b^m, a^n b^m \rangle \mid n, m \in \mathbb{N}\} \cup \{\langle a^n b^m, b^n a^m \rangle \mid n, m \in \mathbb{N}\} \\ c(w) &= \#_a(w) \end{aligned}$$

So the generator is a non-deterministic transducer that takes strings as input that consist of a sequence of *as*, followed by a sequence of *bs*. It either returns the input string unchanged, or it replaces the *as* by *bs* and vice versa. This relation can be constructed as follows (for better readability, I write *a* instead of $\{a\}$ where no confusion is likely to arise):

$$I_{a^*b^*} \cup ((a \times b) \cup (b \times a))^*$$

The constraint *c* can informally be stated as “Avoid *a*”, i.e. every *a* in the output violates *c* once. Thus *c* can also easily be implemented as a finite state automaton. We could for instance identify it with the relation:

$$((b \times e) \cup (a \times c))^*$$

This relation erases all *bs* and replaces every *a* by *c*. Applied to some output *w* of GEN, its output is thus a sequence of *cs* of length $c(w)$.

The relation of optimal input-output pairs induced by this OT-system is the relation $\{\langle a^n b^m, a^n b^m \rangle \mid n \neq m\} \cup \{\langle a^n b^m, b^n a^m \rangle \mid m \leq n\}$. Intersecting the range of this relation with the regular language a^*b^* yields the language $\{a^n b^m \mid n \leq m\}$, which is easily shown to be non-regular.² It thus follows from the closure properties of regular languages and rational relations that the relation defined by this OT-system is not rational, even though both GEN and *c* can be implemented by means of finite state techniques.

More generally, this example shows that gradient constraints transcend the bounds of finite state techniques. Gerdemann & van Noord (2000) show though that it is nevertheless possible to implement certain linguistically useful OT-systems using gradient constraints. To bring this point home, they implement the constraint “Parse!” from Prince & Smolensky (1993) as a regular expression, even though “Parse!” can be violated arbitrarily many times. Their approach is based on the insight that the ordering on outputs that is induced by this constraint is itself a rational relation. In other words, there is a finite state transducer *T* such that every suboptimal candidate (with respect to “Parse!”) can be obtained by applying *T* to some other candidate. For the purpose of illustration, a very simplified version of their implementation is presented here. The candidates to be compared by “Parse!” are different possible syllabifications of a given sequence of phonemes. Syllabification is expressed by means of indexed brackets around each phoneme. There are four kinds of brackets (φ is used as metavariable over phonemes, and the index of the bracket is placed in front of the opening bracket):

- $O[\varphi]$ – onset
- $N[\varphi]$ – nucleus
- $D[\varphi]$ – coda
- $X[\varphi]$ – unparsed

2. If it were regular, the language $(a^*b^* - \{a^n b^m \mid n \leq m\})b = \{a^n b^m \mid 0 < m \leq n\}$ would be regular too, since the regular languages are closed under complement and concatenation. The intersection of this language with the language in question is $\{a^n b^n \mid n > 0\}$, which is known to be non-regular.

The number of violations of “Parse!” equals the number of “unparsed” phonemes in the input string, i.e. the number of occurrences of “X [ϕ]”. Consider the input string BEBOP. Suppose we have to evaluate three possible syllabifications of this string with respect to “Parse!”, namely

- $O[B] N[E] X[B] X[O] X[P]$
- $O[B] N[E] O[B] N[O] X[P]$
- $O[B] X[E] O[B] N[O] X[P]$

Obviously the second candidate is optimal with respect to “Parse!” since it violates this constraint only once while the first and the third candidate violate it twice and three times respectively. Furthermore we can observe that we can obtain both suboptimal candidates from the optimal candidate if we replace at least one, and optionally several, parsed indices (O , N , or D) by X . This kind of replacement is easy to implement by means of a finite state transducer.³ In other words, the relation that holds between two candidates \mathbf{a} and \mathbf{b} iff they are comparable (i.e. they are outputs for the same input) and \mathbf{a} violates the constraint “Parse!” less than \mathbf{b} can be represented as a rational relation.⁴ This fact can be used to construct a filter that eliminates the suboptimal candidates. Let R_P be the rational relation that maps each candidate \mathbf{a} to each competitor \mathbf{b} that violates “Parse!” more often than \mathbf{a} . Let us furthermore suppose for the sake of simplicity that “Parse!” is the only constraint in our OT-system. Composing GEN with R_P is thus the relation that maps each input to each suboptimal output, and $Rg(\text{GEN} \circ R_P)$ is the set of suboptimal outputs. Given this, $\text{GEN} \circ \mathbf{I}_{Rg(\text{GEN}) - Rg(\text{GEN} \circ R_P)}$ is hence the relation that maps each input to its optimal outputs with respect to “Parse!”, and this is a rational relation.

Even though the mentioned authors only consider this and closely related examples, their method can easily be generalized. Let us call a constraint c *rational* with respect to a generator relation R iff there is a rational relation S such that for all candidates x and y it holds that

$$\{\langle x, y \rangle \mid c(x) < c(y)\} \cap (R^\cup \circ R) = S \cap (R^\cup \circ R)$$

Note that two outputs x and y are competitors if they are both outputs of the same input. This means that x and y are competing iff $\langle x, y \rangle \in (R^\cup \circ R)$. A constraint c is thus rational with respect to R iff we can model the ranking that c induces among the competing outputs of R by means of some rational relation S .

3. Let Σ be the alphabet of the OT-system in question, and let

$$R = (O \times X) \cup (N \times X) \cup (D \times X).$$

The rational relation representing this replacement operation would then be

$$(R \cup \mathbf{I}_\Sigma)^* R (R \cup \mathbf{I}_\Sigma)^*.$$

4. Gerdemann and van Noord’s construction is somewhat more involved. They mark every constraint violation with a special violation mark at the position where the violation occurs, and they observe that the violation marked versions of the suboptimal candidates can be obtained from the violation marked version of the optimal candidate by applying a finite state transducer. This difference in implementation has no impact on the general argument though.

In the sequel I will write $rel_R(c)$ for the rational relation that represents a rational constraint c with respect to R . (We will omit the subscript if no confusion is likely to arise.) Following the example of Gerdemann and van Noord, this relation can be used to construct a filter that eliminates all suboptimal candidates from a given set of competitors.

Now suppose R is a rational relation and c a rational constraint with respect to R . Then the *generalized lenient composition* of R with $rel(c)$ (written as “ $R \uparrow rel_R(c)$ ”) relates an input i with an output o iff iRo and among the possible outputs of i under R , o violates c only minimally.

Definition 9 (Generalized lenient composition)

$$R \uparrow S \doteq R \circ \mathbf{I}_{Rg(R)-Rg(RoS)}$$

Clearly the generalized lenient composition of two rational relations is again a rational relation.

A word of caution is in order here though. Consider the following toy OT-system. There are just two inputs, i_1 and i_2 , and two outputs, o_1 and o_2 . The generator relation GEN relates i_1 to both outputs while it maps i_2 just to o_2 . There is one constraint c , which is violated by o_2 , but not by o_1 . So we have

$$\begin{aligned} \text{GEN} &= \{ \langle i_1, o_1 \rangle, \langle i_1, o_2 \rangle, \langle i_2, o_2 \rangle \} \\ c(o_1) &= 0 \\ c(o_2) &= 1 \end{aligned}$$

It is easy to see that the optimal input-output pairing maps i_1 to o_1 and i_2 to o_2 . When we consider the possible outputs of i_1 , o_2 is blocked because it competes with o_1 , and the latter is better according to c . When we consider the possible outputs of i_2 though, o_2 is optimal because there is no competitor that could block it. Now suppose we form the generalized lenient composition of GEN with $rel(c)$. The relation $rel(c)$ is simply $\{ \langle o_1, o_2 \rangle \}$. Composing GEN with $rel(c)$ and forming the range of the result gives us the singleton set $\{o_2\}$. So the filter that we construct from $rel(c)$ excludes o_2 in a global way, and we would wrongly predict that there is no optimal output for i_2 at all. The problem here is that in this scenario, competition and blocking between different outputs depends on the input. In other words, local optimality does not entail global optimality.

Generalized lenient composition only works if this is not the case. To make this point precise, the filter construction is possible if optimality is global in the following sense: If some output o is optimal with respect to some input i , then o is also optimal with respect to the set of all possible outputs, regardless of the input. Formally this reads as follows:

Definition 10 Let R and S be relations. Optimality is global with respect to R and S iff

$$\forall i, o ((iRo \wedge \neg \exists o' (iRo' \wedge o'So)) \rightarrow \neg \exists o' (o' \in Rg(R) \wedge o'So))$$

In Gerdemann and van Noord’s sample construction, optimality is certainly global with respect to GEN and R_p because there the inverse of GEN is functional. It is always

possible to recover a unique input from a given output candidate, just by erasing all brackets. Thus competition only takes place between different bracketings over the same string, and thus local and global optimality coincide.

Generally, if the precondition that optimality be global is fulfilled, generalized lenient composition in fact provides the optimal outputs for each input:

Fact 1 Let R and S be relations such that optimality is global with respect to R and S . Then

$$\langle i, o \rangle \in R \uparrow S \text{ iff } iRo \wedge \neg \exists o' (iRo' \wedge o'So)$$

Proof: We start with the direction from left to right. Suppose $\langle i, o \rangle \in R \uparrow S$. By the definition of generalized lenient composition, this means that iRo and $o \in Rg(R) - Rg(R \circ S)$. So the first conjunct of the right hand side is fulfilled. Now suppose the second conjunct is false, i.e. there is an o' such that iRo' and $o'So$. Since $o \in Rg(R) - Rg(R \circ S)$, there is no o'' such that $o'' \in Rg(R)$ and $o''So$. $o' \in Rg(R)$ by assumption, thus $\neg o'So$. So we derive a contradiction.

We turn to the direction from right to left. Suppose iRo and there is no o' with iRo' and $o'So$. We furthermore assume that optimality is global with respect to R and S . Finally we assume that the left hand side of the biconditional is false, i.e. $\langle i, o \rangle \notin R \uparrow S$. By the definition of generalized lenient composition, this means that $o \notin Rg(R) - Rg(R \circ S)$. Since $o \in Rg(R)$, it follows that $o \in Rg(R \circ S)$. Thus there is an o' with $o' \in Rg(R)$ and $o'So$. Since optimality is global with respect to R and S , there is thus an o'' with iRo'' and $o''So$. This is in contradiction with the assumptions. \dashv

Lenient composition with binary constraints is a special case of the more general notion of generalized lenient composition (provided optimality is global). Suppose the extension of some binary constraint c is the regular language $l(c)$. Then c is a rational constraint, and $rel(c) = (l(c) \times (\Sigma^* - l(c)))$. Then it follows from fact 1 that

$$\langle i, o \rangle \in R \uparrow (l(c) \times (\Sigma^* - l(c))) \text{ iff } iRo \wedge \neg \exists o' (iRo' \wedge o' \in l(c) \wedge o \notin l(c))$$

It furthermore follows from Frank & Satta's theorem that

$$\langle i, o \rangle \in R \uparrow l(c) \text{ iff } iRo \wedge \neg \exists o' (iRo' \wedge o' \in l(c) \wedge o \notin l(c))$$

because the right hand side of this biconditional is just the definition of unidirectional optimality for the OT system $\mathcal{O} = \langle R, c \rangle$. From this one can infer the corollary that

$$R \uparrow l(c) = R \uparrow (l(c) \times (\Sigma^* - l(c)))$$

Since neither Frank & Satta's proof of theorem 1 nor the proof of fact 1 requires the languages and relations involved to be regular/rational, this result holds regardless whether R and c are rational or not. Generalized lenient composition can be used to model optimization with respect to a system of ranked constraints in a way completely analogous to Frank & Satta's construction with ordinary lenient composition:

Theorem 2 Let $\mathcal{O} = \langle \text{GEN}, \mathbb{C} \rangle$ with $\mathbb{C} = \langle c_1, \dots, c_p \rangle$ be an OT-system such that \mathbb{C} solely consists of rational output markedness constraints. Furthermore, let optimality

be global with respect to $\text{GEN} \uparrow \text{rel}(c_1) \dots \uparrow \text{rel}(c_{i-1})$ and $\text{rel}(c_i)$ for all i with $1 \leq i \leq p$. Then $\langle i, o \rangle$ is unidirectionally optimal iff $\langle i, o \rangle \in \text{GEN} \uparrow c_1 \dots \uparrow c_p$.

Proof: Directly from fact 1 and the definition of unidirectional optimality. \dashv

5. Bidirectional OT

In the generative tradition of syntax, phonology and morphology, transformations are taken to be mappings from underlying abstract representations to concrete surface representations. OT researchers usually adopt this perspective too; competition takes place between different possible realizations of some underlying form. In other words, OT usually takes the generation perspective. It is a theory about the optimal realization of a given underlying form.

On a somewhat more abstract level, the OT philosophy can be described by the idea that only the most economical candidates of a given candidate set are legitimate linguistic objects; less economical competitors are blocked. Ranked constraints serve to induce an ordering on the candidates that makes optimization possible. The idea of optimization has a long history in semantics and pragmatics too, and it is suggestive to integrate this tradition into the OT framework. Some caution has to be exerted here though. The generation perspective that is prevalent in phonology and morphology certainly has some plausibility when applied to semantics. Here it amounts to saying that a certain verbalization of a given meaning, though licit, might be blocked by a more economical linguistic form expressing the same meaning. Such effects do in fact occur. A case in point is the well-known phenomenon of “conceptual grinding”, where the name of an animal kind is used to refer to meat of this animal:

- (1) We had chicken for dinner.

However, conceptual grinding is only possible if there is no lexicalized expression for the kind of meat in question:

- (2) a. ?We had pig for dinner.
b. We had pork for dinner.

Arguably, using the lexicalized expression *pork* is a more economical way to refer to meat from pigs than using the noun *pig* in its shifted meaning. Thus (2b) blocks (2a).

On the other hand, there is also a considerable tradition in semantics and pragmatics which assumes that a certain interpretation of a given linguistic form may be blocked by a more coherent alternative interpretation of the same form. In other words, the candidate set for optimization in semantics may also be determined by the parsing perspective, where different interpretations of a given form are compared. A typical example is the behavior of presupposition accommodation. Consider the following two sentences:

- (3) a. If Mary becomes a politician, the president will resign.
 b. If Mary becomes member of [a club]_i, its_i president will resign.

In both examples, the consequent of the conditional contains a definite NP and thus a presupposition trigger. In (3a) the presupposition triggered is *there is a president*, and in (b) *the club in question has a president*. If one assumes that both sentences are uttered out of the blue, these presuppositions must be accommodated. In principle, there are three ways to accommodate this presupposition in (a) (cf. Heim 1990, van der Sandt 1992): local, intermediate and global accommodation. There is agreement in the literature that global accommodation is preferred, thus (3a) is (correctly) predicted to be interpreted as (4):

- (4) There is [a president]_i, and if Mary becomes a politician, he_i will resign.

If global accommodation is impossible as in (3b) (where it would lead to a configuration where the antecedent of the pronoun *it* is not accessible for the pronoun anymore), intermediate accommodation pops up; (3b) comes out as

- (5) If Mary becomes member of a club that has a president, this president will resign.

A concise way to describe this pattern is to assume that the grammar generally admits both kinds of accommodation, but that global accommodation is more economical than intermediate one (which is in turn more economical than local accommodation). So if a construction structurally admits both readings, global accommodation wins and blocks all competing readings.

So it seems that the mapping of linguistic forms to interpretations requires optimization both in the parsing and in the generation direction. This insight is not new, some form of bidirectional optimization has been assumed in the pragmatics literature for quite some time (see for instance Horn 1984 and Levinson 1987). In a series of recent publications, Reinhard Blutner has made the interplay between generation optimization and parsing optimization precise and integrated it into the overall framework of OT (Blutner 1998, Blutner 2001).

So to apply OT to the syntax-semantics interface, both speaker direction and hearer direction should be taken into account. A grammatically licit form-meaning pair—or, more generally, a licit input-output pair— $\langle i, o \rangle$ may be blocked both by a more economical form alternative and a more economical meaning alternative. It should be added that a blocking expression should itself be optimal. So we arrive at the following definition of bidirectional optimality:

Definition 11 (Bidirectional optimality) Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ be an OT-system. Then $\langle i, o \rangle$ is bidirectionally optimal iff

1. $\langle i, o \rangle \in \text{GEN}$,
2. there is no bidirectionally optimal $\langle i', o \rangle \in \text{GEN}$ such that $\langle i', o \rangle <_{\mathcal{O}} \langle i, o \rangle$ and
3. there is no bidirectionally optimal $\langle i, o' \rangle \in \text{GEN}$ such that $\langle i, o' \rangle <_{\mathcal{O}} \langle i, o \rangle$.

So an OT-System induces a ranking of candidates in the usual way. Bidirectional optimality differs from the unidirectional case in that an input-output pair $\langle i, o \rangle$ can be blocked either by a better alternative output o' for i or a better alternative input i' for o .

In Jäger (2001) it is shown that despite its apparent circularity, bidirectional optimality is a well-defined notion. This bidirectional version of optimality is more than just a conjunction of interpretive and productive economy. To illustrate this point, we sketch an application of bidirectionality which is due to Krifka (2000) pertaining to the interpretation of measure phrases. Consider the following two sentences:

- (6) a. The distance between Amsterdam and Vienna is nine hundred sixty eight kilometers.
 b. The distance between Amsterdam and Vienna is one thousand kilometers.

Suppose that the distance between Amsterdam and Vienna is in fact exactly nine hundred sixty eight kilometers. In this situation we would probably consider both sentences as true, even though (6b) is only true in a vague interpretation of *one thousand*. On the other hand, if the distance between the two cities were precisely one thousand kilometers, (6a) would definitely be considered to be false. One can see from this example that simple measure phrases like *one thousand kilometers* can be interpreted both precisely and vaguely, while more complex phrases like *nine hundred sixty eight kilometers* only admit a precise interpretation. (Needless to say that “precise” and “vague” are also vague terms here; we are actually dealing with different degrees of granularity.) Let us assume for the sake of illustration that there is a preference for simple over complex expressions on the one hand, and a preference for vague over precise interpretations on the other hand. (We ignore the issue what constraints would induce exactly this ranking.) If we restrict attention to the two forms *one thousand kilometers* and *nine hundred sixty eight kilometers* and to the two meanings ‘1000 km’ and ‘968 km’, we have to choose from four candidate form-meaning pairs. They are given in the figure below. The arrows between the candidates indicate preferences (this notation is borrowed from Dekker & van Rooy 2000).

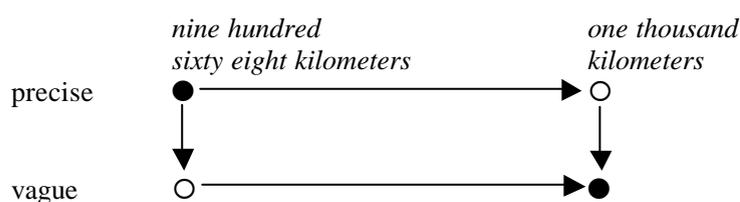


Figure 3

The combination “vague/*one thousand kilometers*” is bidirectionally optimal because there are neither better form alternatives nor better meaning alternatives. Consequently, both the combinations “precise/*one thousand kilometers*” and “vague/*nine hundred sixty eight kilometers*” are blocked and thus not bidirectionally optimal. In other words, there is just one candidate which is both speaker optimal and hearer optimal. However, according to the definition above the combination “precise/*nine hundred sixty eight*

kilometers” is also bidirectionally optimal because both its competing form alternative and its competing meaning alternative are blocked.

6. Finite state bidirectional OT

In Jäger (2001) it is demonstrated that Frank & Satta’s result carries over to the bidirectional case. To gain an intuitive understanding of the construction, let us consider another, slightly more complex example, which illustrates how bidirectional optimality is evaluated in case of a finite GEN. Suppose $\text{GEN} = \{1, 2, 3\} \times \{1, 2, 3\}$, and we have two constraints which both say “Be small!”. One of its instances applies to the input and one to the output. Thus formally we have

- $\mathcal{O} = \langle \text{GEN}, C \rangle$
- $\text{GEN} = \{1, 2, 3\} \times \{1, 2, 3\}$
- $C = \langle c_1, c_2 \rangle$
- $c_1(\langle i, o \rangle) = i$
- $c_2(\langle i, o \rangle) = o$

It follows from the way constraints are evaluated that $\langle i_1, o_1 \rangle <_{\mathcal{O}} \langle i_2, o_2 \rangle$ iff $i_1 \leq i_2$, $o_1 \leq o_2$, and $\langle i_1, o_1 \rangle \neq \langle i_2, o_2 \rangle$. Now obviously $\langle 1, 1 \rangle$ is bidirectionally optimal since both its input and its output obey the constraint in an optimal way. Accordingly, $\langle 1, 2 \rangle$, $\langle 1, 3 \rangle$, $\langle 2, 1 \rangle$ and $\langle 3, 1 \rangle$ are blocked, since they all share a component with a bidirectionally optimal candidate. Among the remaining candidates, $\langle 2, 2 \rangle$ is certainly bidirectionally optimal because all of its competitors in either dimension are known to be blocked. This candidate in turn blocks $\langle 2, 3 \rangle$ and $\langle 3, 2 \rangle$. The only remaining candidate, $\langle 3, 3 \rangle$, is again bidirectionally optimal since all its competitors are blocked.⁵ This example illustrates the general strategy for the finite case: Find the cheapest input-output pairs in the whole of GEN and mark them as bidirectionally optimal. Next mark all candidates that share either the input component or the output component (but not both) with one of these bidirectionally optimal candidates as blocked. If there are candidates left that are neither marked as bidirectionally optimal nor as blocked, repeat the procedure until GEN is exhausted. This strategy can be presented semi-formally as the algorithm in figure 4.

5. Bidirectional optimality thus predicts iconicity: the pairing of cheap inputs with cheap outputs is optimal, but also the pairing of expensive inputs with expensive outputs. See Blutner’s paper for further discussion of this point.

Algorithm

```

OPT = ∅
BLCKD = ∅
while (OPT ∪ BLCKD ≠ GEN) {
  OPT = OPT ∪ {x ∈ GEN - BLCKD |
    ∀y < x : y ∈ OPT ∪ BLCKD};
  BLCKD = BLCKD ∪ {⟨i, o⟩ ∈ GEN - OPT |
    ⟨i', o⟩ ∈ OPT ∨ ⟨i, o'⟩ ∈ OPT};
}
return (OPT);

```

Figure 4: Bidirectional evaluation algorithm for finite GEN

Fact 2 If GEN is finite and $<$ is weakly linear,

1. the algorithm always terminates, and
2. $x \in OPT$ iff x is bidirectionally optimal.

Proof: As abbreviational convention, $x \sqsubset y$ is used if $x < y$, $x = \langle i, o \rangle$ and $y = \langle i', o \rangle$ or $\langle i, o' \rangle$ for some i' or o' .

1. Suppose the algorithm does not terminate. Since OPT stores a set that is never decreased in a *while*-loop, this can only happen if the value of OPT after a reassignment is identical to its value before. Therefore

$$\{x \in \text{GEN} - \text{BLCKD} \mid \forall y < x : y \in \text{OPT} \cup \text{BLCKD}\} \subseteq \text{OPT}$$

By simple set theoretic reasoning, this means that

$$\forall y < x : y \in \text{OPT} \cup \text{BLCKD} \rightarrow x \in \text{OPT} \cup \text{BLCKD}$$

Since “ $<$ ” is well-founded, we can conclude by induction that

$$\text{GEN} \subseteq \text{OPT} \cup \text{BLCKD}$$

So the stop condition of the *while*-loop is fulfilled and the program will terminate.

2. We prove that after each run through the loop, $x \in OPT$ entails that x is optimal and $x \in \text{BLCKD}$ entails that x is not optimal. Suppose that x was added to OPT after the $n + 1$ th run. So x was neither in OPT nor in BLCKD after the n th run. Furthermore, all its predecessors, and thus all its \sqsubset -predecessors, were in OPT or in BLCKD after the n th run. If one of the \sqsubset -predecessors had been in OPT , x had wound up being in BLCKD , so if $y \sqsubset x$, y is not optimal by hypothesis. Thus x is optimal.

Now suppose x was added to BLCKD after the $n + 1$ th run. Then there is an $y \in OPT$ which shares one component with x . By weak linearity of $<$, we know that $y < x \vee x < y \vee x \equiv y$. Since x and y share one component, we can strengthen this to $y \sqsubset x \vee x \sqsubset y \vee x \equiv y$. We can exclude the possibility that $x \sqsubset y$ because we know that y is optimal by induction hypothesis. So suppose that $x \equiv y$, $y \in OPT$; hence all predecessors of y were in $OPT \cup \text{BLCKD}$ after the n th run of the *while*-loop. $x \equiv y$, therefore x and y have the same predecessors, and x should have been

added to OPT at the $n + 1$ th run as well, against assumption. Thus $y \sqsubset x$. It follows from the previous paragraph that y is optimal, hence x is not optimal.

So if $x \in OPT$ after termination of the program, x is optimal. Suppose $x \notin OPT$ after termination. The stop condition of the *while*-loop requires that any x is either in OPT or in $BLCKD$. Thus $x \in BLCKD$, hence x is not optimal. \dashv

If we apply this algorithm to the previous example, we have to go three times through the *while*-loop before the algorithm terminates. The values of the two variables OPT (for “optimal”) and $BLCKD$ (for “blocked”) at the different stages of the program run are given in figure 5.

OPT	$BLCKD$
\emptyset	\emptyset
$\{\langle 1, 1 \rangle\}$	$\{\langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 2, 1 \rangle, \langle 3, 1 \rangle\}$
$\{\langle 1, 1 \rangle, \langle 2, 2 \rangle\}$	$\{\langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 2, 1 \rangle, \langle 3, 1 \rangle, \langle 2, 3 \rangle, \langle 3, 2 \rangle\}$
$\{\langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 3 \rangle\}$	$\{\langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 2, 1 \rangle, \langle 3, 1 \rangle, \langle 2, 3 \rangle, \langle 3, 2 \rangle\}$

Figure 5: Sample run of the algorithm

This algorithm is confined to OT-systems with a finite GEN. The finite state construction in Jäger (2001) is modeled after this algorithm, but it is also applicable to systems with an infinite generator, provided all building blocks are finite state implementable. The first step inside the *while*-loop in the algorithm above amounts to finding the globally optimal input-output pairs in the whole of GEN. As in Frank & Satta’s construction, this global optimization is achieved by successively optimizing with respect to the constraints in CON in the order of their ranking. Jäger’s (2000) construction is also restricted to binary markedness constraints. As a further complication, we have to distinguish between input constraints and output constraints (the former would not make sense in unidirectional OT). The definition of the notion of input markedness constraint is the mirror image of definition 7:

Definition 12 (Input Markedness Constraint) Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ be an OT-system. Constraint c_j is an *input markedness constraint* iff

$$\langle i, o \rangle \in \text{GEN} \wedge \langle i, o' \rangle \in \text{GEN} \rightarrow c_j(\langle i, o \rangle) = c_j(\langle i, o' \rangle)$$

If we want to leniently compose GEN with a binary input markedness constraint, we need a mirror image of Frank & Satta’s lenient composition. Thus backward lenient composition is defined as

$$R \downarrow L \doteq (\mathbf{I}_L \circ R) \cup (R \circ \mathbf{I}_{Rg(R) - Rg(\mathbf{I}_L \circ R)})$$

Furthermore, for reasons that will become clear immediately, in bidirectional optimality it is not sufficient to consider the best outputs for a given input, but we have to look for the best input-output pairs in a global way. Thus bidirectional lenient composition is defined in the following way:

Definition 13 (Bidirectional Lenient Composition) Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ be an OT-system with $C = \langle c_1, \dots, c_p \rangle$, and c_i be a binary markedness constraint.

$$R \uparrow c_i \doteq \begin{cases} R \circ \mathbf{I}_{Rg(\{\epsilon\} \times Rg(R) \uparrow c_i)} & \text{if } c_i \text{ is an output markedness constraint} \\ \mathbf{I}_{\text{Dom}(\text{Dom}(R) \times \{\epsilon\}) \downarrow c_i} \circ R & \text{else} \end{cases}$$

Let us look at this construction in detail. Suppose c_i is an output markedness constraint. $\{\mathbf{e}\} \times Rg(R)$ is a relation that relates the empty string to any possible output of R . Leniently composing this relation with c_i leads to a relation that relates the empty string to those possible outputs of R that are optimal with respect to c_i . So if c_i is fulfilled by some output of R , this relation is just $\{\mathbf{e}\} \times (Rg(R) \cap c_i)$. If no output of R obeys c_i , the relation is just $\{\mathbf{e}\} \times Rg(R)$. In either way, $Rg(\{\{\mathbf{e}\} \times Rg(R) \uparrow c_i)$ is the set of outputs of R that are optimal with respect to c_i . Since c_i only evaluates outputs, $R \uparrow c_i$ is thus the set of $\langle i, o \rangle \in R$ that are optimal with respect to c_i . The same holds *ceteris paribus* if c_i is an input markedness constraint.

Like Frank & Satta's operation, bidirectional lenient composition only makes use of finite state techniques. It follows directly from the closure properties of regular languages and rational relations that $R \uparrow c_i$ is a rational relation provided R is rational and c_i is a regular language.

Note that a certain input-output pair may be evaluated as sub-optimal according to this construction even if it neither shares the input component nor the output component with any better candidate. So while Frank & Satta's lenient composition operates pointwise for each input, bidirectional lenient composition is global.

Fact 3 Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ be an OT-system (with binary markedness constraints only), where $C = \langle c_1, \dots, c_p \rangle$, and let $R \subseteq \text{GEN}$. Then

$$\langle i, o \rangle \in R \uparrow c_1 \dots \uparrow c_p$$

iff $\langle i, o \rangle \in R$, and there are no i', o' with $\langle i', o' \rangle \in R$ and $\langle i', o' \rangle < \langle i, o \rangle$.

The proof of this fact can be found in Jäger (2001).

For simplicity, the notation R^C as is used as shorthand for $\{x \in R \mid \neg \exists y \in R : y <_{\mathcal{O}} x\}$, where \mathcal{O} is some OT-system with C as its constraint component. Intuitively, this operation picks out the globally optimal set of input-output pairs from R . The above fact states that $R^C = R \uparrow c_1 \dots \uparrow c_p$ (where $C = \langle c_1, \dots, c_p \rangle$). Note that R^C is a rational relation if R is rational and all constraints in C are regular languages.

Now reconsider the first command inside the *while*-loop of the algorithm given above:

$$OPT = OPT \cup \{x \in \text{GEN} - \text{BLCKD} \mid \forall y < x : y \in OPT \cup \text{BLCKD}\}$$

This step is equivalent to

$$OPT = OPT \cup \{x \in \text{GEN} - OPT - \text{BLCKD} \mid \forall y < x : y \in OPT \cup \text{BLCKD}\}$$

In other words, this step adds the set $(\text{GEN} - \text{OPT} - \text{BLCKD})^C$ to the variable OPT . Bidirectional lenient composition thus generalizes the first part of the naive algorithm to the general, i.e. possibly infinite case.

Now consider the second component of the *while*-loop:

$$\text{BLCKD} = \text{BLCKD} \cup \{ \langle i, o \rangle \in \text{GEN} - \text{OPT} \mid \langle i, o \rangle \in \text{OPT} \vee \langle i, o' \rangle \in \text{OPT} \}$$

BLCKD only plays an auxiliary role in the algorithm; we basically need it to define the set $\text{GEN} - \text{OPT} - \text{BLCKD}$. This set can be defined directly, using only finite state techniques. It is obvious from the definition of BLCKD that it only contains pairs that share one component with some element from OPT . Now consider the following relation:

$$\mathbf{I}_{\text{Dom}(\text{GEN}) - \text{Dom}(\text{OPT})} \circ \text{GEN} \circ \mathbf{I}_{\text{Rg}(\text{GEN}) - \text{Rg}(\text{OPT})}$$

This relation contains exactly those pairs from GEN that share no component with any element of OPT . In other words, it is just the relation $\text{GEN} - \text{OPT} - \text{BLCKD}$.

This fact can be used to define the values that OPT assumes after each run of the *while*-loop recursively while only using finite state techniques.

$$\text{OPT}_0 = \emptyset$$

$$\text{OPT}_{n+1} = (\mathbf{I}_{\text{Dom}(\text{GEN}) - \text{Dom}(\text{OPT}_n)} \circ \text{GEN} \circ \mathbf{I}_{\text{Rg}(\text{GEN}) - \text{Rg}(\text{OPT}_n)})^C$$

Fact 4 Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ be an OT-system such that all c_i in C are binary markedness constraints. If GEN is a rational relation and all constraints can be represented as regular languages, OPT_n is a rational relation for each n .

Proof: Immediately from the construction of OPT and $(\cdot)^C$ and the closure conditions on regular languages/rational relations. \dashv

The proof that the naive algorithm terminates rests on the fact that GEN is finite. This need not hold for the more general case we are considering now. However, even if GEN is infinite, there are only finitely many possible patterns of constraint violations if we are dealing with OT-systems comprising only binary constraints. This guarantees that the algorithm would also terminate after a bounded number of runs in the more general case, and it can therefore be computed offline.

Lemma 2 Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ be an OT-system with $C = \langle c_1, \dots, c_p \rangle$, where all c_i are binary markedness constraints. Then $\langle i, o \rangle$ is bidirectionally optimal iff $\langle i, o \rangle \in \text{OPT}_{2^p}$.

Proof: See Jäger (2001).

This leads directly to the central result of Jäger (2001), namely the bidirectional counterpart of Frank & Satta's (1998) result.

Theorem 3 Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ be an OT-system with $C = \langle c_1, \dots, c_p \rangle$, where all c_i are binary markedness constraints. Furthermore, let GEN be a rational relation and let all c_i be regular languages. Then the set of bidirectionally optimal elements of GEN is a rational relation.

Proof: Directly from fact 4, lemma 2, and the closure properties of regular languages and rational relations. \dashv

7. Gradient constraints and bidirectionality

The results summarized in the previous section suggest that bidirectional OT is not more complex than unidirectional OT in an automata theoretic sense, despite its considerable conceptual complexity. This is not true though; the construction from section 4 does not carry over to the bidirectional case. To see why, consider the following OT system:

- $\mathcal{O} = \langle \text{GEN}, C \rangle$
- $\text{GEN} = \{ \langle a^i b^j, a^k b^l \rangle \mid i = k \vee j = l \}$
- $C = \langle c_1, c_2 \rangle$
- $c_1(\langle i, o \rangle) = \#_a(i)$
- $c_2(\langle i, o \rangle) = \#_b(o)$

GEN is a rational relation, since it can be constructed in the following way, using only regular building blocks and finite state operations:

$$\mathbf{I}_a^* \mathbf{I}_b^* \circ (((a \times \mathbf{e}) \cup \mathbf{I}_b)^* \circ (\mathbf{e} \times a^*) \mathbf{I}_b^*) \cup ((\mathbf{I}_a \cup (b \times \mathbf{e}))^* \circ \mathbf{I}_a^* (\mathbf{e} \times b^*))$$

We assume two constraints: c_1 : “No a !” applies to the input, and c_2 : “No b !” applies to the output. So the number of violations of c_1 equals the number of a s in the string that is evaluated, and likewise for c_2 . c_1 is ranked higher than c_2 (but this plays no role in the sequel).

Both constraints are rational in the sense defined in section 4. Recall that a constraint is rational iff there is a rational relation that holds between two candidates α and β iff β violates the constraint more severely than α . The domain of GEN consists only of elements of the language a^*b^* . Thus we can transform any input candidate which violates c_1 n times into any other input candidates which violates c_1 m times for $m > n$ by adding a number of a s at the beginning, and optionally adding or erasing some b s at the end. There is thus a rational relation corresponding to this constraint, namely

$$\text{rel}(c_1) : ((\mathbf{e} \times (aa^*)) \mathbf{I}_{a \cup b}^* \circ (\mathbf{I}_a^* (b^* \times b^*)))$$

The same holds *ceteris paribus* for the output constraint c_2 which counts the number of b s in a string. It can be represented by the following rational relation (which is more or less the result of reversing the order of the relation given above and exchanging a and b).

$$\text{rel}(c_2) : (\mathbf{I}_{a \cup b}^* (\mathbf{e} \times bb^*)) \circ ((a^* \times a^*) \mathbf{I}_b^*)$$

Finally, optimality is global with respect to GEN and both $\text{rel}(c_1)$ and $\text{rel}(c_2)$. This follows from the fact that any two outputs of GEN are competitors, i.e. they share at

least one input. (If the two outputs in question are $a^x b^y$ and $a^z b^w$, they share at least the input $a^x b^w$.)

If the construction from the previous section could be generalized to rational gradient constraints in a way similar to Gerdemann & van Noord's generalization of Frank & Satta's result, we would expect the set of bidirectionally optimal input-output pairs for the OT-system to form a rational relation. This is not so, however. To prove this fact, another recursive definition of bidirectional optimality is used that is somewhat more general than the one used in the last section. It is also taken from Jäger (2001).

Definition 14 Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ be an OT-system using only markedness constraints.

$$OPT_0 = \emptyset$$

$$OPT_{\alpha+1} = OPT_{\alpha} \cup [\mathbf{I}_{\text{Dom}(\text{GEN}) - \text{Dom}(OPT_{\alpha})} \circ \text{GEN} \circ \mathbf{I}_{\text{Rg}(\text{GEN}) - \text{Rg}(OPT_{\alpha})}]^C$$

$$OPT_{\alpha} = \bigcup_{\beta < \alpha} OPT_{\beta} \text{ if } \alpha \text{ is a limit ordinal}$$

$$OPT = \bigcup_{\alpha} OPT_{\alpha}$$

This definition generalizes the one used in the previous section to the transfinite case. This is necessary if we deal with gradient constraints since there may be infinitely many different patterns of constraint violations. This cumulative definition coincides with the standard definition of bidirectional optimality:

Fact 5 Let $\mathcal{O} = \langle \text{GEN}, C \rangle$ be an OT-system using only markedness constraints. $\langle i, o \rangle$ is bidirectionally optimal iff $\langle i, o \rangle \in OPT$.

Proof: See Jäger (2001).

We can use this construction to determine the set of bidirectionally optimal candidates from the OT-System given above. The crucial step is the observation that for all finite n :

$$OPT_n = \{\langle a^i b^y, a^z b^i \rangle \mid i < n \wedge (y = i \vee z = i)\} \quad (1)$$

We prove this by complete induction over n . The base case ($n = 0$) is obvious since $OPT_0 = \emptyset$ by definition. Now suppose the claim holds for n . Then, by the definition given above,

$$OPT_{n+1} = OPT_n \cup (\mathbf{I}_{\text{Dom}(\text{GEN}) - \text{Dom}(OPT_n)} \circ \text{GEN} \circ \mathbf{I}_{\text{Rg}(\text{GEN}) - \text{Rg}(OPT_n)})^C$$

Now $\text{Dom}(OPT_n) = \{a^i b^k \mid i < n\}$, thus $\text{Dom}(\text{GEN}) - \text{Dom}(OPT_n) = \{a^i b^k \mid i \geq n\}$. Likewise, $\text{Rg}(\text{GEN}) - \text{Rg}(OPT_n) = \{a^k b^i \mid i \geq n\}$. We will use the abbreviation

$$R_0 = \mathbf{I}_{\text{Dom}(\text{GEN}) - \text{Dom}(OPT_n)} \circ \text{GEN} \circ \mathbf{I}_{\text{Rg}(\text{GEN}) - \text{Rg}(OPT_n)}$$

We thus have

$$OPT_{n+1} = OPT_n \cup R_0^C$$

$$R_0 = \{\langle a^x b^y, a^z b^w \rangle \mid x, w \geq n \wedge (x = z \vee y = w)\}$$

To determine R_0^C , we have to bidirectionally optimize R_0 with respect to c_1 and c_2 . The effect of c_1 is to minimize the number of as in the input. So the globally optimal elements of R_0 with respect to c_1 are those with the minimal number of as in the input, i.e.:

$$\{\langle a^n b^y, a^z b^w \rangle \mid (n = z \vee y = w) \wedge w \geq n\}$$

Optimizing this relation with respect to c_2 minimizes the number of bs in the output, so we get

$$R_0^C = \{\langle a^n b^y, a^z b^n \rangle \mid n = z \vee n = y\}$$

Together with the induction hypothesis, we now have

$$OPT_{n+1} = \{\langle a^i b^y, a^z b^i \rangle \mid i \leq n \wedge (y = i \vee z = i)\}$$

This completes the proof of (1).

From this we can infer that

$$OPT_\omega = \{\langle a^n b^y, a^z b^n \rangle \mid y = n \vee z = n\}$$

We will now demonstrate that $OPT = OPT_\omega$. By the recursive definition of OPT_α , it holds that

$$OPT_{\omega+1} = OPT_\omega \cup (\mathbf{I}_{Dom(GEN) - Dom(OPT_\omega)} \circ GEN \circ \mathbf{I}_{Rg(GEN) - Rg(OPT_\omega)})^C$$

It obviously holds that $Dom(OPT_\omega) = Dom(GEN)$ and $Rg(OPT_\omega) = Rg(GEN)$. Therefore $Dom(GEN) - Dom(OPT_\omega) = Rg(GEN) - Rg(OPT_\omega) = \emptyset$. Furthermore, $\emptyset^C = \emptyset$. We therefore conclude that

$$OPT_{\omega+1} = OPT_\omega$$

The same argument can be applied to every transfinite ordinal number α . So if $\alpha \geq \omega$, it holds that

$$OPT_\alpha = OPT_\omega$$

Hence

$$OPT = OPT_\omega = \{\langle a^n b^y, a^z b^n \rangle \mid y = n \vee z = n\}$$

Now suppose this relation were rational. Then the result of composing it with a rational relation would again be a rational relation. The relation $\{\langle a^n, a^n \rangle \mid n > 0\}$ is certainly rational, because it can be defined as:

$$\mathbf{I}_{aa^*}$$

Thus $\mathbf{I}_{aa^*} \circ OPT$ should be rational as well. This is the relation $\{\langle a^n, a^n b^n \rangle \mid n > 0\}$. The range of this relation is known to be a non-regular language, thus it cannot be a rational relation, and hence OPT cannot be rational either.

This simple example demonstrates that a finite state implementation of gradient constraints along the lines of Gerdemann & van Noord (2000) is not possible in the context of bidirectional OT. In other words, the optimization strategy of bidirectional OT is intrinsically more complex than its unidirectional counterpart.

At the present point I am unable to pair this negative result with a positive one. There is no known upper limit for the complexity of bidirectionally evaluated OT-systems consisting of finite state building blocks—even though our example suggests that we stay within the limits of simple pushdown automata (and the corresponding transducers). A precise characterization has to be left for future research.

On the other hand, our example is suggestive from the point of view of a theoretical linguist. The optimal input-output relation in the example displays a recursive structure that is missing in the generator relation. We might thus speculate that certain recursive aspects of natural language are not due to a (computationally) powerful generator, but that they emerge from a process of bidirectional optimization. Future work has to show whether it is possible to come up with linguistically realistic OT-systems that display a behavior similar to our abstract example.

8. Conclusion

In this paper some issues pertaining to the computational complexity of optimality were discussed, both in its standard (i.e. unidirectional) and its bidirectional variety (where Bidirectional OT is interpreted in the sense of Blutner 2001). Two noteworthy results on unidirectional OT from the literature were reviewed. Frank & Satta (1998) propose a general way to implement unidirectional optimization by means of finite state techniques. Their construction is confined to OT systems that only employ binary markedness constraints. Gerdemann & van Noord (2000) give a finite state implementation for an OT system using gradient constraints. The present paper generalized this construction and showed that it is always applicable (a) if the OT system in question only uses markedness constraints, and (b) the notions of global and local optimality defined by this system coincide.

The present paper furthermore reviewed the main result from Jäger (2001), which gives a finite state construction for bidirectional optimality which is based on Frank & Satta's construction and is applicable under the same restrictions. Finally it was shown that a similar extrapolation of Gerdemann and van Noord's construction to bidirectionality is not possible. Given that all the mentioned constructions impose fairly strong constraints on the OT systems to be implemented, one might wonder whether these results are of much linguistic importance. As far as the unidirectional case is concerned, the restrictions that we have to assume are no serious obstacle for practical applications though. Most contemporary work in OT tacitly or explicitly adopts the premises of Correspondence Theory (McCarthy & Prince 1995) according to which the outputs of GEN are not just surface structures but pairs of underlying and surface structures, enriched with correspondence information. This means that the input is always uniquely recoverable from the output.

In other words, GEN is one-to-many; the inverse of GEN is a function. In such OT systems, all constraints are markedness constraints, and global optimality always coincides with local optimality. So practical applications of the complexity results solely

depend on the issue whether particular versions of GEN and of the constraints involved can be implemented as rational relations, which has to be decided on a case-by-case basis. The practical usage of the the results on bidirectional OT is less straightforward. First of all, the existing applications of bidirectionality mainly deal with syntax, semantics and pragmatics, and in these areas finite state automata have a too limited generative capacity to realistically model linguistic phenomena. The only module of grammar where both finite state techniques and bidirectionality are promising tools seems to be morphology. It has to be acknowledged however that a many-to-one generator does not make much sense in a bidirectional setting, because then bidirectional and unidirectional OT would coincide. So for a non-trivial usage of bidirectionality, one has to give up the assumptions of Correspondence Theory. If this is done, the restriction to markedness constraints turns out to be a severe limitation. Not surprisingly, the only serious attempt to apply bidirectionality in morphology—namely Wunderlich (2001a)—makes crucial use of faithfulness constraints.

Nonetheless the complexity results on bidirectional OT that were presented in this article are significant from a theoretical point of view. The fact that bidirectional optimization that uses gradient (markedness) constraints transcends the bounds of finite state techniques while unidirectional optimization does not indicates that bidirectional optimization is intrinsically more complex than its unidirectional counterpart. The obvious question that is to be addressed in future work is how the complexity impact of bidirectional optimization can be characterized in a precise way. Besides the repercussions of this negative result for the appropriate division of labor between the generator and the optimization procedure deserves to be investigated.

References

- Beaver, David (2000). The optimization of discourse. Manuscript, Stanford.
- Blutner, Reinhard (1998). Lexical pragmatics. *Journal of Semantics* 15, 115–162.
- Blutner, Reinhard (2001). Some aspects of optimality in natural language interpretation. *Journal of Semantics* 17, 189–216.
- Boersma, Paul and Bruce Hayes (2001). Empirical tests of the gradual learning algorithm. *Linguistic Inquiry* 32, 45–86.
- Dekker, Paul and Robert van Rooy (2000). Bi-directional optimality theory: An application of game theory. *Journal of Semantics* 17, 217–242.
- Egg, Markus (1999). Deriving and resolving ambiguities in *wieder*-sentences. In *Proceedings of the Twelfth Amsterdam Colloquium*, Paul Dekker (ed.), 109–114. University of Amsterdam.
- Frank, Robert and Giorgio Satta (1998). Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24 (1), 307–315.
- Gerdemann, Dale and Gertjan van Noord (2000). Approximation and exactness in finite state optimality theory. In *Finite State Phonology. Proceedings of SIGPHON 2000*, Jason Eisner, Lauri Karttunen and Alain Thériault (eds.). ACL.

- Haspelmath, Martin (1999). Optimality and diachronic adaptation. *Zeitschrift für Sprachwissenschaft* 18, 180–205.
- Hayes, Bruce (1999). Phonological acquisition in optimality theory: The early stages. manuscript, UCLA.
- Heim, Irene (1990). On the projection problem for presuppositions. In *Pragmatics*, Steven Davis (ed.), 397–405. Oxford: Oxford University Press.
- Horn, Lawrence (1984). Towards a new taxonomy for pragmatic inference: Q-based and R-based implicatures. In *Meaning, Form, and Use in Context*, Deborah Schiffrin (ed.), 11–42. Washington: Georgetown University Press.
- Jäger, Gerhard (2001). Some notes on the formal properties of bidirectional Optimality Theory. to appear in *Journal of Logic, Language and Information*.
- Jäger, Gerhard and Reinhard Blutner (2000). Against lexical decomposition in syntax. In *Studies in Optimality Theory*, Reinhard Blutner and Gerhard Jäger (eds.), 5–29. University of Potsdam.
- Joshi, Aravind, K. Vijay-Shanker and David Weir (1990). The convergence of mildly context-sensitive grammar formalisms. In *Processing of Linguistic Structure*, Tom Wasow and Peter Sells (eds.), 31–81. Cambridge (Mass.): MIT Press.
- Kager, René (1999). *Optimality Theory*. Cambridge: Cambridge University Press.
- Kaplan, Ronald and Martin Kay (1981). Phonological rules and finite-state transducers. In *Linguistic Society of America Meeting Handbook*. New York: LSA.
- Karttunen, Lauri (1998). The proper treatment of optimality in computational phonology. Manuscript. Xerox Research Centre Europe.
- Krifka, Manfred (2000). Precise and vague interpretations of measure terms: An easy piece in bidirectional Optimality Theory. Paper presented at Sinn & Bedeutung V, University of Amsterdam.
- Lee, Hanjung (1999). The emergence of the unmarked order. Manuscript, Stanford University.
- Levinson, Stephen C. (1987). Pragmatics and the grammar of anaphora. *Journal of Linguistics* 23, 379–434.
- McCarthy, John and Alan Prince (1995). Faithfulness and reduplicative identity. In *Papers in Optimality Theory*, Jill Beckman, Laura Walsh Dickey and Suzanne Urbanczyk (eds.), 259–384. University of Massachusetts Occasional Papers in Linguistics 18, Amherst: GLSA.
- Prince, Alan and Paul Smolensky (1993). Optimality theory: Constraint interaction in generative grammar. Technical Report TR-2, Rutgers University Cognitive Science Center, New Brunswick, NJ.
- Roche, Emmanuel and Yves Schabes (1997). Introduction. In *Finite-State Language Processing*, Emmanuel Roche and Yves Schabes (eds.), 1–65. Cambridge, Ma.: MIT Press.
- Tesar, Bruce and Paul Smolensky (2001). *Learnability in Optimality Theory*. Cambridge, Ma.: The MIT Press.
- van der Sandt, Rob (1992). Presupposition projection as anaphora resolution. *Journal of Semantics* 9, 333–377.

-
- von Stechow, Arnim (2000). How are results represented and modified? Remarks on Jäger's and Blutner's anti-decomposition. Manuscript, University of Tübingen.
- Wilson, Colin (2001). Bidirectional optimization and the theory of anaphora. In *Optimality-theoretic Syntax*, Géraldine Legendre, Jane Grimshaw and Sten Vikner (eds.). Cambridge, Ma.: MIT Press.
- Wunderlich, Dieter (2001a). Bidirectional OT in morphology. Ms., University of Düsseldorf.
- Wunderlich, Dieter (2001b). How gaps and substitutions can become optimal: the pronominal affix paradigms of Yimas. *Transactions of the Philological Society* 99, 315–366.
- Zeevat, Henk (1999). Explaining presupposition triggers. In *Proceedings of the Twelfth Amsterdam Colloquium*, Paul Dekker (ed.), 19–24. University of Amsterdam.