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

Gerhard Jäger

Utrecht Institute of Linguistics (OTS)

`Gerhard.Jaeger@let.uu.nl`

`http://www.let.uu.nl/~Gerhard.Jaeger/personal`

1 Finite state OT: Frank and Satta (1998)

Optimality Theory in the sense of Prince and Smolensky 1993 is computationally very expensive in the general case. It can be shown that the set of optimal candidates for a given generator **GEN** and a set of constraints **CON** may be undecidable even if both **GEN** and all constraints in **CON** are recursive. Under certain general conditions, however, optimization does not even exceed the limits of finite state techniques. Frank and Satta (1998) show that the optimal input-output relation is rational if (a) **GEN** is a rational relation and (b) all constraints in **CON** are binary markedness constraints that have a regular language as extension. The proof of this fact makes crucial use of an operation called "conditional intersection" (Karttunen 1998 calls it "lenient composition"), which, in the finite state calculus from Kaplan and Kay (1994), can be defined as

$$\texttt{R lc L} \doteq \{\texttt{R} \circ \texttt{L}, \texttt{(domain(R) - domain(R} \circ \texttt{L))} \circ \texttt{R}\}$$

The optimal input-output relation can be defined by successively leniently composing **GEN** with the (extensions of the) constraints in **CON** in the order of their strength.

2 Bidirectional OT

Blutner (2000) introduces a notion of **bidirectional OT** which rests on the intuition that an input-output pair competes both with alternative outputs for the given input (as in standard OT) and with alternative inputs for the given output. Bidirectional OT has been successfully applied to the analysis of quite a few phenomena, mainly in the area of semantics and pragmatics. Existing applications of bidirectional OT make heavy use of "gradient" or "counting" constraints, i.e. constraints that can be violated more than once. Typical examples are "Do not accommodate!" or "Avoid focus!". The present paper explores the impact of gradient constraints on the automata theoretic complexity of both unidirectional and bidirectional OT. Its main result states that for a given class of Gen-relations and (gradient) constraints, bidirectional OT is more complex than unidirectional OT. While this class of OT-systems is clearly too limited to model the intended applications of bidirectionality in semantics and pragmatics—it is restricted to Gen-Relations and constraints that can be modeled by finite automata—it nevertheless illuminates the intrinsic complexity of bidirectionality, and it may serve as starting point for more general investigations into the complexity of OT.

Jäger (2000) defines bidirectional OT in the way given below (which is equivalent to Blutner's original definition under very general conditions).

Definition 1 (OT-System):

1. An OT-system is a pair $\mathcal{O} = \langle \mathbf{GEN}, C \rangle$, where **GEN** is a relation, and $C = \langle c_1, \ldots, c_p \rangle, p \in \mathbb{N}$ is a linearly ordered sequence of functions from **GEN** to $\mathbb{N}$.

2. Let $a, b \in \mathbf{GEN}$. $a <_{\mathcal{O}} b$ iff there is an i with $1 \leq i \leq p$ such that $c_i(a) < c_i(b)$ and for all $j < i : c_j(a) = c_j(b)$.

Definition 2 (Bidirectional Optimality): An input-output pair $\langle i, o \rangle$ is 2-optimal iff

1. $\langle i, o \rangle \in \mathbf{GEN}$,
2. there is no 2-optimal $\langle i', o \rangle$ such that $\langle i', o \rangle < \langle i, o \rangle$.
3. there is no 2-optimal $\langle i, o' \rangle$ such that $\langle i, o' \rangle < \langle i, o \rangle$.

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 (2000) it is shown that despite its apparent circularity, 2-optimality is a well-defined notion.

3 Finite state bidirectional OT

It is furthermore shown in Jäger (2000) that Frank and Satta's result carries over to the bidirectional case. To gain an intuitive understanding of the construction, let us consider how 2-optimality is evaluated in case of a finite $\mathbf{GEN}$. Suppose $\mathbf{GEN} = \{1,2,3\} \times \{1,2,3\}$, and we have two constraints which both say "Be small!". One of its instance applies to the input and one to the output. So we have $\langle i_1, o_1 \rangle < \langle i_2, o_2 \rangle$ 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 2-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 2-optimal candidate. Among the remaining candidates, $\langle 2, 2 \rangle$ is certainly 2-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 2-optimal since all its competitors are blocked.[1] This example illustrates the general strategy for the finite case: Find the cheapest input-output pairs in the whole of $\mathbf{GEN}$ and mark them as 2-optimal. Next mark all candidates that share either the input component or the output component (but not both) with one of these 2-optimal candidates as blocked. If there are candidates left that are neither marked as 2-optimal nor as blocked, repeat the procedure until $\mathbf{GEN}$ is exhausted.

[1] 2-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.

The finite state construction in Jäger (2000) is modeled after this algorithm. The first step amounts to finding the globally optimal input-output pairs in the whole of $\mathbf{GEN}$. As in Frank and Satta's construction, this global optimization is achieved by successively optimizing with respect to the constraints in $\mathbf{CON}$ in the order of their ranking. Jäger's 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).

Definition 3 (Bidirectional lenient composition): Let $\mathcal{O} = \langle \mathbf{GEN}, \mathbf{CON} \rangle$ be an OT-system and c_i be a binary markedness constraint.

$$\mathtt{R\ blc\ c}_i \doteq \begin{cases} \mathtt{R} \circ \mathtt{range(([]\ x\ range(R))\ lc\ c}_i\mathtt{)} \\ \quad \text{if } \mathtt{c}_i \text{ is an output constraint} \\ \mathtt{range(([]\ x\ domain(R))\ lc\ c}_i\mathtt{)} \circ \mathtt{R} \\ \quad \text{if } \mathtt{c}_i \text{ is an input constraint} \end{cases}$$

The set of globally optimal input-output pairs with respect to a system of ranked constraints $\mathbf{CON} = \langle c_1, \ldots, c_n \rangle$ can now be defined as (where `blc` is assumed to be left-associative):

Definition 4 (Global optimization):

$$\mathtt{glop(R,}\mathbf{CON}\mathtt{)} \doteq \mathtt{R\ blc\ c}_1\ \mathtt{blc} \cdots \mathtt{blc\ c}_n$$

The recursive aspect of the naive algorithm given above can be modeled by means of the recursive definition given in figure 1 on the following page.

Suppose after n round of marking candidates either as 2-optimal or as blocked, a certain set of pairs $\mathtt{2opt(}n\mathtt{,R,}\mathbf{CON}\mathtt{)}$ is known to be 2-optimal. Restricting $\mathbf{GEN}$ to those inputs that do not occur in the domain of $\mathtt{2opt(}n\mathtt{,R,}\mathbf{CON}\mathtt{)}$ and to those outputs that do not occur in the range of $\mathtt{2opt(}n\mathtt{,R,}\mathbf{CON}\mathtt{)}$ excludes all input-output pairs that are either in $\mathtt{2opt(}n\mathtt{,R,}\mathbf{CON}\mathtt{)}$ or blocked by some element of $\mathtt{2opt(}n\mathtt{,R,}\mathbf{CON}\mathtt{)}$. The remaining relation is thus the set of pairs that are neither known to be 2-optimal nor known to be blocked at the current state. Applying the `glop` operation to this set delivers the optimal elements of this set. It is easy to see that $\mathtt{2opt(}n+1\mathtt{,R,}\mathbf{CON}\mathtt{)}$ is a rational relation if `R` and $\mathtt{2opt(}n\mathtt{,R,}\mathbf{CON}\mathtt{)}$ are and all constraints are regular, and $\mathtt{2opt(0,R,}\mathbf{CON}\mathtt{)}$ is certainly

Definition 5:

```
2opt(0,R,CON)    ≐  {}
2opt(n+1,R,CON)  ≐  {2opt(n,R,CON),
                     glop(
                     ~domain(2opt(n,R,CON)) ∘ R ∘ ~range(2opt(n,R,CON)),
                     CON)}
```

Fig. 1: Recursive definition of 2-optimality

rational. Thus if **GEN** is rational and all constraints in **CON** are regular markedness constraints, `2opt(`n`,`**GEN**`,`**CON**`)` is a rational relation for arbitrary n. In Jäger (2000) it is proved that the set of 2-optimal candidates is `2opt(`2^i`,`**GEN**`,`**CON**`)`, where i is the number of constraints in **CON**. Bidirectional optimality can thus be modeled by means of finite state techniques provided we are only dealing with binary markedness constraints and all components of the OT system in question are finite state objects.

4 Gradient constraints in unidirectional OT

Both Frank and Satta's and Jäger's construction are restricted to binary constraints. Karttunen (1998) shows how Frank and Satta's result can be generalized to counting constraints which have an upper bound on the number of constraint violations, and the same trick can be applied to the bidirectional case. You simply have to replace a constraint c which admits at most n violations by $n + 1$ binary constraints of the form "Don't violate c at all!", "Violate c at most once", ..., "Violate c at most n times!". No matter how these new constraints are ranked with respect to each other, they will induce the same ranking of candidates as the original counting constraint. This technique is not applicable though if there is no upper bound for the number of violations. Gerdemann and van Noord (2000) present an alternative approach to the finite state modeling of gradient constraints. They implement the constraint "Parse!" from Prince and 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.

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* iff there is a rational relation `R` such that for all candidates x and y it holds that

$$c(x) < c(y) \iff x\mathtt{R}y$$

Intuitively, the relation `R` represents the ranking that is induced by the constraint c. In the sequel we will write $rel(c)$ for the rational relation that represents a rational constraint c.

Suppose `R` is a rational relation and c a rational constraint. Then the *generalized lenient composition* of `R` with $rel(c)$ (written as "`R glc` $rel(c)$") relates an input i with an output o iff $i\mathtt{R}o$ and among the possible outputs of i under `R`, o violates c only minimally.

Definition 6 (Generalized lenient composition):

```
R glc S ≐ R ∘ ~range(range(R) ∘ S)
```

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

Lenient composition with binary constraints is a special case of this more general notion. Suppose the extension of some binary constraint c is the regular language $l(c)$. Then c is a rational constraint, and $rel(c) = l(c)$ `x ~`$l(c)$. As the reader may convince himself, it generally holds that

$$\text{R lc L} = \text{R glc (L x \textasciitilde L)}$$

The considerations from this section suggest a general recipe for implementing OT-systems with gradient constraints as regular expression: Try to represent gradient constraints as rational relations and use generalized lenient composition!

5 Gradient constraints and bidirectionality

The result summarized in section 3 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 the previous section does not carry over to the bidirectional case. To see why, consider the following OT system: **GEN** is given by the regular expression

$$\texttt{[a*,b*]} \circ \{\texttt{\{a x [], b\}*} \circ \texttt{[[] x a*, b*]}, \texttt{\{a, b x []\}} \circ \texttt{[a*, [] x b*]}\}$$

It defines the relation $\{\langle a^i b^j, a^k b^l\rangle | i = k \vee j = l\}$. We assume two constraint: 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 as 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). The corresponding relations are (if we restrict the domain to `[a*,b*]`, which is sufficient for the given **GEN**):

$$\begin{aligned} rel(c_1)&: \quad \texttt{[[] x a+, ?*]} \\ rel(c_2)&: \quad \texttt{[?*, [] x b+]} \end{aligned}$$

The set of 2-optimal input-output pairs with respect to this OT system is the set $\{\langle a^i b^k, a^i b^i\rangle | i, k \geq 0\} \cup \{\langle a^i b^i, a^k b^i\rangle | i, k \geq 0\}$. The proof of this fact is given in the appendix. It is easy to see that this relation cannot be rational. If it were, the image of the regular language a^+ under this relation would be regular too, but this image is the non-regular $\{a^n b^n | n > 0\}$.
To conclude this section, due to the recursive definition of bidirectional optimality, bidirectional optimization with respect to gradient constraints involves an aspect of counting (which is missing in the unidirectional case) that cannot be modeled by means of finite state techniques. Thus despite the results from Jäger (2000) that suggest the contrary, bidirectional OT has a higher automata theoretic complexity than unidirectional OT.

Appendix

To establish that the set of 2-optimal pairs for the OT-system given in the last section is $\{\langle a^i b^y, a^z b^i\rangle | y = i \vee z = i\}$, we make use of the fact that for arbitrary n, $\texttt{2opt}(n, \mathbf{GEN}, \mathbf{CON})$ is a subset of the set of 2-optimal elements of **GEN** with respect to **CON** (which is proven in Jäger 2000). Let us abbreviate $\texttt{2opt}(n, \mathbf{GEN}, \mathbf{CON})$ for the **GEN** and **CON** given above as $\texttt{2opt}(n)$. We first prove by complete induction over n that for all n:

$$\text{(1)} \qquad \texttt{2opt}(n) = \{\langle a^i b^y, a^z b^i\rangle | i < n \wedge (y = i \vee z = i)\}$$

The base case ($n = 0$) is obvious since $\texttt{2opt}(0) = \emptyset$ by definition. Now suppose the claim holds for n. Then by definition 5,

$$\begin{aligned} \texttt{2opt}(n+1) = {} & \texttt{2opt(n)} \cup \\ & \texttt{glop(\~{}domain(2opt(n))} \\ & \circ\, \mathbf{GEN} \circ \texttt{\~{}range(2opt(n))}, \\ & \mathbf{CON}) \end{aligned}$$

Now $\texttt{domain}(\texttt{2opt}(n)) = \{\langle a^i b^k\rangle | i < n\}$, thus $\texttt{\~{}domain}(\texttt{2opt}(n)) = \{\langle a^i b^k\rangle | i \geq n\}$. Likewise, $\texttt{\~{}range}(\texttt{2opt}(n)) = \{\langle a^k b^i\rangle | i \geq n\}$. Thus

$$\texttt{\~{}domain}(\texttt{2opt}(n)) \circ \mathbf{GEN} \circ \texttt{\~{}range}(\texttt{2opt}(n))$$

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

Let us call this relation R_0. To evaluate $\texttt{glop}(R_0, \mathbf{CON})$, we have to replace `lc` in the definition of `blc` by `glc` (since we are dealing with gradient constraints). Given this, we get

$$R_0 \;\texttt{blc}\; rel(c_1) = \{\langle a^n b^y, a^z b^w\rangle | w \geq n \wedge (z = n \vee y = w)\}$$

(The effect of c_1 is minimizing the number of as in the input.) Let us call this relation R_1. Applying constraint c_2 gives

$$R_1 \;\texttt{blc}\; rel(c_2) = \{\langle a^n b^y, a^z b^n\rangle | z = n \vee y = n)\}$$

(c_2 minimizes the number of bs in the output.) Putting the pieces together, this entails that $\texttt{2opt}(n+1) = \{\langle a^i b^y, a^z, b^i\rangle | i < n + 1 \wedge (y = i \vee z = i)\}$, which completes the proof of (1). It remains to be shown that $\texttt{2opt}(\omega + \alpha) = \texttt{2opt}(\omega)$ for arbitrary ordinals α, where $\texttt{2opt}(\omega) = \bigcup_{n<\omega} \texttt{2opt}(n)$. That this is so follows from the fact that

$$\texttt{\~{}domain}(\texttt{2opt}(\omega)) \circ \mathbf{GEN} \circ \texttt{\~{}range}(\texttt{2opt}(\omega)) = \emptyset$$

hence $\texttt{2opt}(\omega + 1) = \texttt{2opt}(\omega)$, and likewise for all other transfinite ordinals. ⊣

References

Blutner, R. (2000). Some aspects of optimality in natural language interpretation. to appear in *Journal of Semantics*.

Frank, R. and G. Satta (1998). Optimality theory and the generative complexity of constraint violability. *Computational Linguistics*, **24**(1):307–315.

Gerdemann, D. and G. van Noord (2000). Approximation and exactness in finite state optimality theory. In J. Eisner, L. Karttunen, and A. Thériault, eds., *Finite State Phonology. Proceedings of SIGPHON 2000*. ACL.

Jäger, G. (2000). Some notes on the formal properties of bidirectional optimality theory. In R. Blutner and G. Jäger, eds., *Studies in Optimality Theory*, pp. 41–63. University of Potsdam.

Kaplan, R. and M. Kay (1994). Regular models of phonological rule systems. *Computational Linguistics*, **20**(3):331–379.

Karttunen, L. (1998). The proper treatment of optimality in computational phonology. manuscript. Xerox Research Centre Europe.

Prince, A. and P. Smolensky (1993). Optimality theory: Constraint interaction in generative grammar. Technical Report TR-2, Rutgers University Cognitive Science Center, New Brunswick, NJ.