Journal of Logic, Language and Informationl manuscript No.
(will be inserted by the editor)

A note on the GLA's choice of the current loser from the perspective of factorizability

Giorgio Magri

Received: date / Accepted: date

Abstract Boersma's (1997, 1998) *Gradual Learning Algorithm* (GLA) performs a sequence of slight re-rankings of the constraint set triggered by mistakes on the incoming stream of data. Data consist of underlying forms paired with the corresponding winner forms. At each iteration, the algorithm needs to complete the current data pair with a corresponding *loser* form. Tesar and Smolensky (1998) suggest that this current loser should be set equal to the winner predicted by the current ranking. This paper develops a new argument for Tesar and Smolensky's proposal, based on the GLA's *factorizability*. The underlying typology often encodes non-interacting phonological processes, so that it *factorizes* into smaller typologies that encode a single process each. The GLA should be able to take advantage of this factorizability, in the sense that a run of the algorithm on the original typology should *factorize* into independent runs on the factor typologies. Factorizability of the GLA is guaranteed provided the current loser is set equal to the current prediction, providing new support for Tesar and Smolensky's proposal.

Keywords Optimality Theory · Gradual Learning Algorithm · learnability

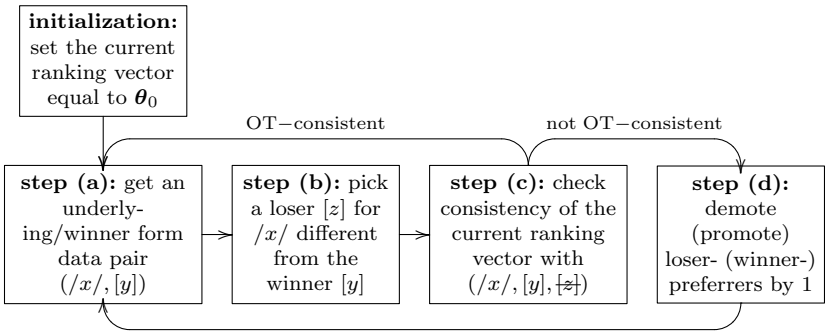
1 introduction

The *Gradual Learning Algorithm* (henceforth: GLA) is an error-driven learning algorithm for *Optimality Theory* (henceforth: OT; Prince and Smolensky 2004, Kager 1999) developed in Boersma (1997, 1998). The (non-stochastic variant of the) GLA is described in (1). The algorithm entertains a current *ranking vector* $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$, namely an assignment of numerical *ranking values* $\theta_1, \ldots, \theta_n$ to the n phonological constraints $C_1, \ldots, C_n$. This current ranking

G. Magri
SFL UMR 7023 (CNRS, University of Paris 8)
Tel.: +39 (0)6 52 85 32 32
E-mail: magrigrg@gmail.com

vector is initialized to an assigned initial ranking vector $\boldsymbol{\theta}_0$. And it is updated over time by looping through the four steps (1a)-(1d).

(1)



At step (1a), the algorithm receives a piece of data, consisting of an underlying form $/x/$ paired with the corresponding winner form $[y]$. At step (1b), the algorithm picks another candidate surface form $[z]$ (different from the winner $[y]$) for this underlying form $/x/$, that represents a *loser* candidate. At step (1c), the algorithm checks whether the current underlying/winner/loser form triplet[1] $(/x/, [y], \text{[}\not{z}\text{]})$ assembled at steps (1a) and (1b) is *OT-consistent* with its current ranking vector, according to condition (2). A constraint is *winner-* (respectively, *loser-*) *preferring* relative to the underlying/winner/loser form triplet $(/x/, [y], \text{[}\not{z}\text{]})$ iff it assigns strictly more (respectively, strictly less) violations to the loser mapping $(/x/, \text{[}\not{z}\text{]})$ than to the winner mapping $(/x/, [y])$; see Prince (2002) for relevant discussion.

(2) There exists a constraint that is winner-preferring relative to the underlying/winner/loser form triplet $(/x/, [y], \text{[}\not{z}\text{]})$ and has a current ranking value strictly larger than the ranking value of any constraint that is instead loser-preferring.

If consistency holds, nothing happens: the algorithm loops back to step (1a) and waits for another piece of data. Otherwise, the algorithm needs to slightly update its current ranking vector in response to its failure on the current triplet $(/x/, [y], \text{[}\not{z}\text{]})$. This current failure provides heuristic evidence that loser-preferring constraints are currently ranked too high while winner-preferring constraints are currently ranked too low. The GLA thus decreases the current ranking values of loser-preferring constraints by a fixed amount, say 1 for concreteness; and furthermore increases the current ranking values of winner-preferring constraints by that same amount, as in (1d). The GLA then loops back to step (1a), waits for more data and starts all over again.

The GLA is trained on a single piece of data at the time and does not keep track of previously seen data, so that is does not impose unrealistic memory requirements. Furthermore, a run of the GLA describes a sequence of intermediate ranking vectors that offers a straightforward tool for modeling

[1] As a mnemonic, I strike out the third item "$\text{[}\not{z}\text{]}$" of the triplet, to signal that $[z]$ is the intended loser candidate, while $[y]$ is the intended winner.

observed child acquisition paths. Indeed, the algorithm has been reported to have good modeling properties; see Boersma and Levelt (2000), Curtin and Zuraw (2002), and Boersma and Hayes (2001).

In this squib, I focus on the proper implementation of step (1b), namely on the proper subroutine the GLA should use in order to select the current loser form [~~z~~]. Tesar and Smolensky (1998, p. 246) make an influential proposal on this issue in the passage quoted below.

(3) "A [winner y] is received, [corresponding to] an input x. It is natural for the learner to compute her own [predicted winner y_{pred}] for [the input x], optimal with respect to her current [ranking]. If the learner's [predicted winner y_{pred}] is different from the target [winner y], learning should be possible; otherwise, it isn't. This is because if the target [winner y] equals the learner's [predicted winner y_{pred}], then [y] is already optimal according to [the current ranking] [...] and no learning is possible. On the other hand, if the target [winner y] is not the [winner y_{pred} currently predicted by the learner], then [the latter] is suboptimal [...], and the [current ranking] needs to be modified [...]. In order for a loser [candidate z] to be informative when paired with the winner [candidate y], the Harmony of the loser [candidate z] (according to the current [ranking]) must be greater than the Harmony [of the intended winner y]; only then will [an update] occur to render [the intended winner y] more harmonic than the loser [z]. The obvious choice for this loser [z] is [the winner y_{pred} currently predicted by the learner]: it is of maximum Harmony according to the current ranking, and if any competitor to the winner has higher Harmony according to the current ranking, then [y_{pred}] must."

Based on this reasoning, Tesar and Smolensky as well as Boersma define the subroutine for the choice of the loser form as follows:

(4) At step (1b), the GLA sets the current loser [z] equal to the winner [y_{pred}] predicted by the current ranking vector for the current underlying form /x/

Statement (4) needs to be qualified as follows. In the passage quoted in (3), Tesar and Smolensky assume that the learner represents its current hypothesis on the target adult grammar in the form of a *ranking*, as it is standard in OT. The GLA instead represents its current hypothesis in the form of a numerical *ranking vector*. If a ranking vector has pairwise distinct ranking values, then it can be identified with the unique ranking that ranks a constraint above another constraint iff the ranking value of the former constraint is larger than the ranking value of the latter. If a ranking vector has instead two or more identical components, then it can be identified with multiple rankings, that resolve the ties in different ways and are thus called *refinements* of the given ranking vector. To illustrate, the ranking vector in (5a) admits the unique refinement $C_1 \gg C_2 \gg C_3$, while the ranking vector in (5b) admits the two

refinements $C_1 \gg C_2 \gg C_3$ and $C_1 \gg C_3 \gg C_2$, as there are two different ways to break the tie between the two ranking values of C_2 and C_3.

(5) a. $\begin{matrix} C_1 \\ C_2 \\ C_3 \end{matrix}\begin{bmatrix} 100 \\ 70 \\ 50 \end{bmatrix}$ b. $\begin{matrix} C_1 \\ C_2 \\ C_3 \end{matrix}\begin{bmatrix} 100 \\ 50 \\ 50 \end{bmatrix}$

A ranking vector is consistent with an underlying/winner/loser form triplet according to condition (2) provided each one of its refinements is consistent with that triplet in the usual OT sense. From now on, the expression "the winner y_{pred} predicted by the current ranking vector" in (4) is to be intended as a short hand for the more careful expression "the winner y_{pred} predicted by an *arbitrary* refinement of the current ranking vector" in the usual OT sense (see Boersma 2009 for relevant discussion).

Tesar and Smolensky's reasoning quoted in (3) has two steps. In the general case, some of the loser candidates $[z]$ will give rise to an underlying/winner/loser form triplet $(/x/, [y], \cancel{[z]})$ that is consistent with the current ranking vector while some other loser candidates will give rise to a triplet that is instead inconsistent. The latter case is more advantageous than the former case. In fact, an *inconsistent* triplet prompts the algorithm to update its current ranking vector and thus to learn from the current piece of data $(x/, [y])$. On the contrary, a *consistent* triplet just keeps the algorithm waiting for more data, impeding any learning from the current piece of data. As a *first step* of their reasoning, Tesar and Smolensky thus suggest that the algorithm should choose the current loser $[z]$ in such a way that the corresponding underlying/winner/loser form triplet $(/x/, [y], \cancel{[z]})$ is inconsistent with the current ranking vector, whenever possible. It turns out that is possible only in case the candidate $[y_{\text{pred}}]$ predicted by the current ranking vector is different from the current intended winner $[y]$. As a *second step* of their reasoning, Tesar and Smolensky thus refine their initial suggestion, setting the current loser $[z]$ always equal to the current prediction $[y_{\text{pred}}]$. The first step of Tesar and Smolensky's reasoning seems to me compelling. But the second step doesn't. Plausibly, computing the current optimum $[y_{\text{pred}}]$ is computationally more demanding than just picking an arbitrary inconsistent loser. And Tesar and Smolensky provide no justification for this extra computational cost. Unless such a learning theoretic justification can be provided, their proposal (4) for the choice of the current loser thus remains unwarranted.

This paper provides such a justification, based on the notion of *factorizability*. It is standard practice in phonological analyses to focus on a restricted set of phonological processes. This practice is motivated by the intuition that interactions among different phonological processes are in some sense sparse. For instance, vowel round harmony and final obstruent devoicing are largely independent processes that can be studied separately. Section 2 straightforwardly formalizes this intuition into an explicit notion of *factorizable* OT typologies, namely typologies that describe independent processes and can therefore be factorized into smaller, independent *factor typologies*. Section 3 argues that the GLA should be able to take advantage of the factorizability of the underlying

OT typology. In the sense that a run of the GLA on the original factorizable typology should *factorize* into independent runs on the multiple factor typologies. Section 4 then looks at conditions that ensure factorizability of the GLA. It turns out that factorizability requires the current underlying/winner/loser form triplets to be constructed properly. In particular, setting the current loser $[z]$ equal to the current prediction $[y_{\text{pred}}]$ at step (1b) suffices to ensure factorizability, thus providing the needed argument in support of Tesar and Smolensky's proposal (4).

2 Factorizable typologies

It is standard practice in phonological analyses to focus on a restricted set of phonological processes. This practice is motivated by the intuition that interactions among different phonological processes are in some sense sparse. This Section formalizes this intuition into the notion of *factorizable* OT-typologies.

To start with an example, consider the set of (underlying and surface) forms in (6a), that consists of low, mid and high vowels, both rounded and unrounded. Consider next the constraint set in (6b), that consists of the three faithfulness constraints C_1, C_3 and C_5 for height and roundness as well as the markedness constraints C_2, C_4 and C_6 that punish round, high and mid vowels, respectively.

(6) a. $\left\{ \begin{array}{ccc} \text{y} & \text{ø} & \text{œ} \\ \text{i} & \text{e} & \text{æ} \end{array} \right\}$

b. $\left\{ \begin{array}{lll} C_1 = \textsc{Ident}[\textsc{round}] & C_3 = \textsc{Ident}[\textsc{high}] & C_5 = \textsc{Ident}[\textsc{low}] \\ C_2 = *[+\textsc{round}] & C_4 = *[+\textsc{high}] & C_6 = *[-\textsc{high},\ -\textsc{low}] \end{array} \right\}$

Crucially, there is no constraint that targets at the same time height and roundness, that therefore do not interact in the OT typology (6). This fact has important consequences. Indeed, split up the original constraint set (6b) into the subset (7b) consisting of the constraints for roundness and the subset (8b) consisting of the constraints for height.

(7) a. $\left\{ \begin{array}{ccc} \text{y} & \text{ø} & \text{œ} \\ \text{i} & \text{e} & \text{æ} \end{array} \right\}$ b. $\left\{ \begin{array}{l} C_1 = \textsc{Ident}[\textsc{round}] \\ C_2 = *[+\textsc{round}] \end{array} \right\}$

(8) a. $\left\{ \begin{array}{ccc} \text{y} & \text{ø} & \text{œ} \\ \text{i} & \text{e} & \text{æ} \end{array} \right\}$ b. $\left\{ \begin{array}{ll} C_3 = \textsc{Ident}[\textsc{high}] & C_5 = \textsc{Ident}[\textsc{low}] \\ C_4 = *[+\textsc{high}] & C_6 = *[-\textsc{high},\ -\textsc{low}] \end{array} \right\}$

Since there are no constraints that target both roundness and height, these two constraint subsets (7b) and (8b) *partition* the original constraint set (6b). Starting from the original typology (6), I have thus constructed the two *factor* typologies (7) and (8).

The ranking (9a) over the original constraint set (6b) is consistent with the mapping of the underlying form /y/ into the winner surface form [i], in the usual OT sense. This ranking (9a) over the original constraint set (6b) induces the *factor rankings* (9b) and (9c) over the constraint subsets (7b) and (8b).

(9) a. ∗[+ROUND] / ∗[−HIGH, −LOW] / IDENT[ROUND] / IDENT[HIGH] / IDENT[LOW] / ∗[+HIGH]

b. ∗[+ROUND] / IDENT[ROUND]

c. ∗[−HIGH, −LOW] / IDENT[HIGH] / IDENT[LOW] / ∗[+HIGH]

Because of lack of interaction between roundness and height, each of these two factor rankings is by itself consistent with the mapping of /y/ into [i]. As the latter property holds for any mapping and any ranking, I say that the original typology (6) *factorizes* into the two factor typologies (7) and (8).

A ranking generates a *language*, namely the corresponding set of licit surface forms. Factorizability has of course straightforward consequences at the level of languages. Consider for instance the language (10a), that bans mid vowels and round vowels. This language belongs to the original typology (6), as it corresponds for instance to the constraint ranking (9a). The languages corresponding to the two factor rankings (9b) and (9c) in the two factor typologies (7) and (8) are provided in (10b) and (10c), and will be called the *factor languages*. They consist, respectively, of all unrounded vowels and of all non-mid vowels.

(10) a. { i, æ } b. { i, e, æ } c. { y, œ / i, æ }

Because of lack of interaction between roundness and height, the original language (10a) coincides with the intersection of the two factor languages (10b) and (10c) and can therefore be reconstructed from the factor languages.

These considerations can be generalized straightforwardly, as follows. Consider an OT *typology* $\tau = (\mathcal{X}, \mathcal{Y}, Gen, \mathcal{C})$, namely a set of *underlying forms* $\mathcal{X}$, a set of *surface forms* $\mathcal{Y}$, a *generator function* Gen and a *constraint set* $\mathcal{C}$. Consider a partition $\mathcal{C}', \mathcal{C}''$ of the constraint set $\mathcal{C}$, namely a pair $\mathcal{C}', \mathcal{C}''$ of disjoint subsets of $\mathcal{C}$ whose union coincides with $\mathcal{C}$.[2] The two typologies $\tau' = (\mathcal{X}, \mathcal{Y}, Gen, \mathcal{C}')$ and $\tau'' = (\mathcal{X}, \mathcal{Y}, Gen, \mathcal{C}'')$ corresponding to these two constraint subsets are called the *factor typologies* of the *original* typology τ. Furthermore, the restrictions $\gg'$ and $\gg''$ to the constraint subsets $\mathcal{C}'$ and $\mathcal{C}''$ of an arbitrary ranking $\gg$ on $\mathcal{C}$ are called the *factor rankings* of the *original* ranking $\gg$. The original ranking $\gg$ provides three types of information: information on the relative ranking of constraints in $\mathcal{C}'$; information on the relative ranking of constraints in $\mathcal{C}''$; and information on the relative ranking of a constraint in $\mathcal{C}'$ and one in $\mathcal{C}''$. The two factor rankings $\gg'$ and $\gg''$ preserve the first two pieces of information but lose the latter piece of information. A

[2] I limit myself to partitions of the constraint set into *two* subsets $\mathcal{C}', \mathcal{C}''$, but the reasoning straightforwardly extends to partitions into an arbitrary finite number of subsets.

typology is *factorizable* provided that the latter piece of information is indeed irrelevant, in the sense that the following condition holds: for every underlying form $/x/ \in \mathcal{X}$, every candidate $[y] \in Gen(x)$, and every ranking $\gg$, if the original ranking $\gg$ on $\mathcal{C}$ is consistent with the mapping of $/x/$ into $[y]$ according to the original typology τ, then both factor rankings $\gg'$ and $\gg''$ on $\mathcal{C}', \mathcal{C}''$ are consistent with that mapping as well.[3]

Let $L = L(\tau, \gg)$ be the language in the original typology τ corresponding to the original ranking $\gg$, namely the range of the corresponding OT-grammar. Factorizability entails that this language L coincides with the intersection of the two languages $L' = L(\tau', \gg')$ and $L'' = L(\tau'', \gg'')$ corresponding to the two factor rankings $\gg'$ and $\gg''$ in the two factor typologies τ' and τ'', as stated in (11).

(11) $L = L' \cap L''$

The two languages $L' = L(\tau', \gg')$ and $L'' = L(\tau'', \gg'')$ are called the two *factor languages* of the *original* language $L = L(\tau, \gg)$.

Feature based typologies provide a simple example of factorizable typologies. Let me say that an OT typology τ is *based* on N partial binary phonological features $\varphi_1, \ldots, \varphi_N$ provided that forms can be identified with vectors or sequences of vectors of values of these features; and the constraints are all stated in terms of these features as well. A feature based typology factors relative to a partition $\mathcal{C}', \mathcal{C}''$ of the constraint set provided that the features can be split into two disjoint subsets Φ', Φ'' in such a way that constraints in $\mathcal{C}'$ (respectively, $\mathcal{C}''$) only target features in Φ' (respectively, Φ''). For instance, the OT typology (6) is based on the three features $\varphi_1 = \textsc{round}$, $\varphi_2 = \textsc{high}$, and $\varphi_3 = \textsc{low}$. That typology indeed factors relative to the partition of the constraint set into (7b) and (8b), that corresponds to the partition of the features into $\Phi' = \{\varphi_1\}$ and $\Phi'' = \{\varphi_2, \varphi_3\}$.

3 Why the GLA should factorize

This Section shows how to take advantage of the factorizability of the original typology in the analysis of the GLA, by factorizing the challenging analysis of the algorithm on the original typology with a large constraint set into simpler analyses on non-interacting factor typologies with smaller constraint sets each.

To start with an example, let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_6)$ denote the generic ranking vector for the constraint set $\{C_1, \ldots, C_6\}$ described in (6b). A possible update at the tth iteration in a run of the GLA on the OT typology (6) is described in (12). Let me explain this update in some detail.

[3] The reverse implication always, trivially holds: if the two factor rankings $\gg'$ and $\gg''$ are each consistent with a certain mapping, then the original ranking $\gg$ is consistent with that mapping as well.

(12)

	current vector $\boldsymbol{\theta}_{t-1}$		updated vector $\boldsymbol{\theta}_t$
C_1=IDENT[ROUND]	20		**21**
C_2=*[+ROUND]	30		**29**
C_3=IDEND[HIGH]	17	(/y/, [y], ~~[i]~~) →	17
C_4=*[+HIGH]	15		15
C_5=IDEND[LOW]	5		5
C_6=*[−HIGH, −LOW]	3		3

At step (1a), the GLA is fed the piece of data (/y/, [y]), whereby the underlying form /y/ is faithfully mapped to itself.[4] The current ranking vector $\boldsymbol{\theta}_{t-1}$ (assembled at the end of the preceding $(t-1)$th iteration) admits a unique refinement, namely the ranking $C_2 \gg C_1 \gg C_3 \gg C_4 \gg C_5 \gg C_6$. The winner predicted by this ranking $\gg$ for the underlying form /y/ is [i], as shown by the tableau in (13).

(13)

/y/		C_2 = *[+ROUND]	C_1 = IDENT[ROUND]	C_3 = IDENT[HIGH]	C_4 = *[+HIGH]	C_5 = IDENT[LOW]	C_6 = *[-HIGH, -LOW]
a.	[y]	*!			*		
b.	[ø]	*!		*			*
c.	[œ]	*!		*		*	
d. ☞	[i]		*		*		
e.	[e]		*	*!			*
f.	[æ]		*	*!		*	

Following Tesar and Smolensky's proposal (4), assume that at step (1b) the GLA sets the current loser equal to the incorrect predicted winner [i], thus effectively assembling the current underlying/winner/loser form triplet (/y/, [y], ~~[i]~~). At step (1d), the GLA thus demotes the loser-preferrer $C_2 = *[+\text{ROUND}]$ by 1 and promotes the winner-preferrer $C_1 = \text{IDENT[ROUND]}$ by 1, leading to the updated ranking vector $\boldsymbol{\theta}_t$. How can a sequence of updates like (12) performed by the GLA in the original run be analyzed? Here is a way to go.

Let $\boldsymbol{\phi}$ denote the generic ranking vector for the factor constraint set $\{C_1, C_2\}$ in (7b). Furthermore, let $\boldsymbol{\varphi}$ denote the generic ranking vector for the factor constraint set $\{C_3, C_4, C_5, C_6\}$ in (8b). Two possible updates at the tth iteration in two runs of the GLA on the two factor typologies (7) and (8) are described in (14a) and (14b).

[4] The assumption that the underlying and winner forms coincide is made here for illustrative purposes only. It does not bear in any way on the main point of the paper, which is indeed independent of any assumptions on the underlying/winner data pairs fed to the algorithm.

(14) a.

	current vector $\boldsymbol{\phi}_{t-1}$		updated vector $\boldsymbol{\phi}_t$
C_1=IDENT[ROUND] C_2=*[+ROUND]	$\begin{bmatrix} 20 \\ 30 \end{bmatrix}$	$\xrightarrow{(/\text{y}/, [\text{y}], \cancel{[\text{i}]})}$	$\begin{bmatrix} \mathbf{21} \\ \mathbf{29} \end{bmatrix}$

b.

	current vector $\boldsymbol{\varphi}_{t-1}$		updated vector $\boldsymbol{\varphi}_t$
C_3=IDEND[HIGH] C_4=*[+HIGH] C_5=IDEND[LOW] C_6=*[−HIGH, −LOW]	$\begin{bmatrix} 17 \\ 15 \\ 5 \\ 3 \end{bmatrix}$	$\xrightarrow{(/\text{y}/, [\text{y}], \cancel{[\text{i}]})}$	$\begin{bmatrix} 17 \\ 15 \\ 5 \\ 3 \end{bmatrix}$

Assume that in both updates, the GLA receives the data pair (/y/, [y]) at step (1a), whereby the underlying form /y/ is faithfully mapped into itself. Assume furthermore that the GLA picks the loser form [i] at step (1b), thus again assembling the underlying/winner/loser form triplet (/y/, [y], $\cancel{[\text{i}]}$). Within the factor constraint set $\{C_1, C_2\}$ in (7b), $C_2 = *[+\text{ROUND}]$ is loser-preferring while $C_1 = \text{IDENT}[\text{ROUND}]$ is winner-preferring. The former outranks the latter according to the current ranking vector $\boldsymbol{\phi}_{t-1}$ in (14a). The loser-preferrer $C_2 = *[+\text{ROUND}]$ is thus demoted by 1 and the winner-preferrer $C_1 = \text{IDENT}[\text{ROUND}]$ is promoted by 1, yielding $\boldsymbol{\phi}_t$. Within the factor constraint set $\{C_3, C_4, C_5, C_6\}$ in (8b), there are no loser-preferrers. Thus, no update is performed in (14b), and $\boldsymbol{\varphi}_t$ coincides with $\boldsymbol{\varphi}_{t-1}$.

The current ranking vector $\boldsymbol{\theta}_{t-1}$ in (12) can be assembled from the two current ranking vectors $\boldsymbol{\phi}_{t-1}$ and $\boldsymbol{\varphi}_{t-1}$ in (14): the ranking values assigned by $\boldsymbol{\theta}_{t-1}$ to constraints in $\{C_1, C_2\}$ and $\{C_3, C_4, C_5, C_6\}$ coincide with the ranking values assigned by $\boldsymbol{\phi}_{t-1}$ and $\boldsymbol{\varphi}_{t-1}$, respectively. Crucially, the update preserves this property. In fact, also the updated ranking vector $\boldsymbol{\theta}_t$ in (12) can be assembled from the two corresponding ranking vectors $\boldsymbol{\phi}_t$ and $\boldsymbol{\varphi}_t$ in (14): the ranking values assigned by $\boldsymbol{\theta}_t$ again coincide with the ranking values assigned by $\boldsymbol{\phi}_t$ and $\boldsymbol{\varphi}_t$. I will thus say that the original iteration (12) *factorizes* into the two *factor iterations* (14). If each iteration in a run of the GLA on the original typology (6) factorizes, that run can be reconstructed from the two corresponding factor runs on the two factor typologies (7) and (8). The "difficult" problem of analyzing the run of the GLA on the original typology (6) with a large number of constraints can thus be broken down into the two "easier" problems of analyzing the algorithm on the two factor typologies (7) and (8) with a smaller number of constraints.

These considerations can be generalized straightforwardly, as follows. Consider a run (15) of the GLA on a typology $\tau = (\mathcal{X}, \mathcal{Y}, Gen, \mathcal{C})$ starting from an initial ranking vector $\boldsymbol{\theta}_0$. Here, I denote compactly by $\mathbf{d}_t$ the underlying/winner/loser form data triplet constructed by the GLA at the tth iteration of its loop (1a)-(1d). Furthermore, I denote by $\boldsymbol{\theta}_t$ the current ranking vector entertained by the GLA at the end of the tth iteration. If the current ranking vector $\boldsymbol{\theta}_{t-1}$ at the end of the preceding $(t-1)$th iteration is consistent with the current data triplet $\mathbf{d}_t$ considered at the tth iteration, then no update is

performed at the tth iteration and the two current ranking vectors $\boldsymbol{\theta}_{t-1}$ and $\boldsymbol{\theta}_t$ entertained at the end of the two consecutive iterations coincide. Otherwise, $\boldsymbol{\theta}_t$ is obtained from $\boldsymbol{\theta}_{t-1}$ by demoting (promoting) by 1 those constraints in $\mathcal{C}$ that are loser- (winner-) preferring relative to $\mathbf{d}_t$.

$$(15)\quad \boldsymbol{\theta}_0 \xrightarrow{\mathbf{d}_1,\mathcal{C}} \boldsymbol{\theta}_1 \xrightarrow{\mathbf{d}_2,\mathcal{C}} \boldsymbol{\theta}_2 \xrightarrow{\mathbf{d}_3,\mathcal{C}} \dots \xrightarrow{\mathbf{d}_{t-1},\mathcal{C}} \boldsymbol{\theta}_{t-1} \xrightarrow{\mathbf{d}_t,\mathcal{C}} \boldsymbol{\theta}_t \xrightarrow{\mathbf{d}_{t+1},\mathcal{C}} \dots$$

Consider a partition $\mathcal{C}', \mathcal{C}''$ of the constraint set $\mathcal{C}$, namely $\mathcal{C}'$ and $\mathcal{C}''$ are disjoint subsets of $\mathcal{C}$ whose union coincides with $\mathcal{C}$. Let me denote by $\boldsymbol{\phi}$ and $\boldsymbol{\varphi}$ the generic ranking vectors for the constraint subsets $\mathcal{C}'$ and $\mathcal{C}''$, respectively. Consider the runs (16a) and (16b) of the GLA of the two corresponding factor typologies τ' and τ''. At each iteration, the GLA constructs the same underlying/winner/loser form triplet $\mathbf{d}_t$ constructed in the original run (15). If the current ranking vector $\boldsymbol{\phi}_{t-1}$ (respectively, $\boldsymbol{\varphi}_{t-1}$) entertained at the end of the preceding $(t-1)$th iteration is consistent with the triplet $\mathbf{d}_t$ considered at the tth iteration, then no update is performed. Otherwise, $\boldsymbol{\phi}_t$ (respectively, $\boldsymbol{\varphi}_t$) is obtained from $\boldsymbol{\phi}_{t-1}$ (respectively, $\boldsymbol{\varphi}_{t-1}$) by demoting (promoting) by 1 those constraints in $\mathcal{C}'$ (respectively, $\mathcal{C}''$) that are loser- (winner-) preferring relative to the current triplet $\mathbf{d}_t$.

$$(16)\quad \text{a.}\quad \boldsymbol{\phi}_0 \xrightarrow{\mathbf{d}_1,\mathcal{C}'} \boldsymbol{\phi}_1 \xrightarrow{\mathbf{d}_2,\mathcal{C}'} \boldsymbol{\phi}_2 \xrightarrow{\mathbf{d}_3,\mathcal{C}'} \dots \xrightarrow{\mathbf{d}_{t-1},\mathcal{C}'} \boldsymbol{\phi}_{t-1} \xrightarrow{\mathbf{d}_t,\mathcal{C}'} \boldsymbol{\phi}_t \xrightarrow{\mathbf{d}_{t+1},\mathcal{C}'} \dots$$

$$\text{b.}\quad \boldsymbol{\varphi}_0 \xrightarrow{\mathbf{d}_1,\mathcal{C}''} \boldsymbol{\varphi}_1 \xrightarrow{\mathbf{d}_2,\mathcal{C}''} \boldsymbol{\varphi}_2 \xrightarrow{\mathbf{d}_3,\mathcal{C}''} \dots \xrightarrow{\mathbf{d}_{t-1},\mathcal{C}''} \boldsymbol{\varphi}_{t-1} \xrightarrow{\mathbf{d}_t,\mathcal{C}''} \boldsymbol{\varphi}_t \xrightarrow{\mathbf{d}_{t+1},\mathcal{C}''} \dots$$

For any ranking vector $\boldsymbol{\theta}$ on the original constraint set $\mathcal{C}$, let me denote by $\boldsymbol{\theta}'$ and $\boldsymbol{\theta}''$ the sub-vector of ranking values assigned by $\boldsymbol{\theta}$ to the constraints in $\mathcal{C}'$ and $\mathcal{C}''$, respectively. Let me say that the *original run* (15) of the GLA *factorizes* into the two *factor runs* (16) provided that the two factor runs "mimic" the original run, in the following sense. Assume that the original run (15) starts from an initial ranking vector $\boldsymbol{\theta}_0$ that assigns to the constraints in $\mathcal{C}'$ and $\mathcal{C}''$ the same ranking values as the initial ranking vectors $\boldsymbol{\phi}_0$ and $\boldsymbol{\varphi}_0$ of the two factor runs (16), as stated in (17a). Then, the current ranking vector $\boldsymbol{\theta}_t$ at any time t in the original run (15) assigns to the constraints in $\mathcal{C}'$ and $\mathcal{C}''$ the same ranking values as the ranking vectors $\boldsymbol{\phi}_t$ and $\boldsymbol{\varphi}_t$ entertained at that same time t in the two factor runs (16), as stated in (17b).

$$(17)\quad \text{a.}\quad \boldsymbol{\theta}'_0 = \boldsymbol{\phi}_0 \text{ and } \boldsymbol{\theta}''_0 = \boldsymbol{\varphi}_0;$$

$$\text{b.}\quad \boldsymbol{\theta}'_t = \boldsymbol{\phi}_t \text{ and } \boldsymbol{\theta}''_t = \boldsymbol{\varphi}_t \text{ at any time } t.$$

In other words, the two factor runs completely describe the original run, as the current ranking vector in the original run can be assembled from the two corresponding current ranking vectors in the two factor runs.

Factorizability plays a crucial role in the analysis of the GLA. Here is a concrete way to appreciate this point. Child language acquisition is *gradual*: the target adult grammar is approached through a path of intermediate stages. The GLA describes a sequence of ranking vectors that can be matched with

child acquisition paths, thus providing a tool to model child acquisition gradualness; see for instance Boersma and Levelt (2000) and Curtin and Zuraw (2002). Formal analyses of the sequences of ranking vectors predicted by the GLA are thus needed in order to provide solid ground for this modeling enterprise. For instance, a macroscopic property of child acquisition paths is *restrictiveness*: the intermediate acquisition stages entertained by the child tend to correspond to a more restrictive phonotactics than the target adult one. In order for the GLA to provide a proper model of child acquisition, it thus needs to enforce restrictiveness. This modeling requirement can be formalized by requiring condition (18) to hold at any time t in the run considered. Here, $L(\tau, \gg_t)$ is the language in the original typology τ corresponding to an arbitrary refinement $\gg_t$ of the current ranking vector $\boldsymbol{\theta}_t$ entertained by the GLA at the tth iteration. Condition (18) requires this current language predicted by the GLA to be always smaller than (i.e. a subset of) the target language L_{target} the algorithm is being trained on.

$$L(\tau, \gg_t) \subseteq L_{\text{target}} \tag{18}$$

Formal analyses of the sequences of ranking vectors predicted by the GLA should thus in particular be able to assess how well the algorithm enforces this restrictiveness condition (18). Yet, these analyses are really only viable when the underlying constraint set $\mathcal{C}$ is somewhat small (see for instance Magri 2012). Factorizability now enters the scene. Consider a partition $\mathcal{C}', \mathcal{C}''$ of the original large constraint set $\mathcal{C}$. As the factor constraint subsets $\mathcal{C}'$ and $\mathcal{C}''$ are smaller than the original constraint set $\mathcal{C}$, formal analyses of the GLA on the corresponding factor typologies τ' and τ'' should be possible, or at least easier. Suppose that the GLA can indeed be shown to be restrictive when run on each factor typology τ', τ'' separately. What can be said about the GLA's restrictiveness on the original typology τ? The answer is provided by the following Theorem, that illustrates the importance of the GLA's factorizability.

Theorem 1 *If both the underlying typology and the GLA factorize relative to a partition of the constraint set, then restrictiveness of the GLA on the two corresponding factor typologies entails restrictiveness of the GLA on the original typology.* ■

Proof Let $\boldsymbol{\theta}_t$ be the current ranking vector entertained by the GLA at the end of the tth iteration of the original run (15) on the original constraint set $\mathcal{C}$. Let $\gg_t$ be an arbitrary refinement of $\boldsymbol{\theta}_t$. Let $L(\tau, \gg_t)$ be the language in the original typology τ corresponding to this ranking $\gg_t$. Consider a partition $\mathcal{C}', \mathcal{C}''$ of the constraint set $\mathcal{C}$ and let $\gg'_t$ and $\gg''_t$ be the two factor rankings induced by this refinement $\gg_t$ on the two constraint subsets $\mathcal{C}'$ and $\mathcal{C}''$. Let $L(\tau', \gg'_t)$ and $L(\tau'', \gg''_t)$ be the two factor languages of the language $L(\tau, \gg_t)$, namely the languages in the two factor typologies τ' and τ'' corresponding to the factor rankings $\gg'_t$ and $\gg''_t$, respectively. The hypothesis that the original typology τ factorizes means that the language $L(\tau, \gg_t)$ coincides with the intersection of the two corresponding factor languages $L(\tau', \gg'_t)$ and $L(\tau'', \gg''_t)$, as stated in (19).

(19) *Step I:*
$L(\tau, \gg_t) = L(\tau', \gg'_t) \cap L(\tau'', \gg''_t)$.

Let $\boldsymbol{\theta}'_t, \boldsymbol{\theta}''_t$ be the sub-vectors of ranking values assigned by $\boldsymbol{\theta}_t$ to the constraints in $\mathcal{C}', \mathcal{C}''$. Since $\gg_t$ is a refinement of $\boldsymbol{\theta}_t$ and $\gg'_t, \gg''_t$ are the restrictions of $\gg_t$ to $\mathcal{C}', \mathcal{C}''$, then $\gg'_t, \gg''_t$ are refinements of $\boldsymbol{\theta}'_t, \boldsymbol{\theta}''_t$, as stated in (20).

(20) *Step II:*
the rankings $\gg'_t, \gg''_t$ are refinements of the ranking vectors $\boldsymbol{\theta}'_t, \boldsymbol{\theta}''_t$.

Let $\boldsymbol{\phi}_t, \boldsymbol{\varphi}_t$ be the ranking vectors on $\mathcal{C}', \mathcal{C}''$ entertained by the GLA at the end of the tth iteration of the two corresponding factor runs (16) on the factor typologies τ', τ''. The hypothesis that the GLA factorizes means that $\boldsymbol{\theta}'_t = \boldsymbol{\phi}_t$ and $\boldsymbol{\theta}''_t = \boldsymbol{\varphi}_t$. Fact (20) thus entails that $\gg'_t, \gg''_t$ are also refinements of $\boldsymbol{\phi}_t, \boldsymbol{\varphi}_t$, as stated in (21).

(21) *Step III:*
the rankings $\gg'_t, \gg''_t$ are refinements of the ranking vectors $\boldsymbol{\phi}_t, \boldsymbol{\varphi}_t$.

By (21), the languages $L(\tau', \gg'_t)$ and $L(\tau'', \gg''_t)$ are also the languages entertained at the tth iteration in the two factor runs. Suppose that the GLA is trained on a target language L_{target} in the original run. Then in the two factor runs it is trained on the two factor languages L'_{target} and L''_{target}. The hypothesis that the GLA satisfies restrictiveness on the two factor typologies thus entails that $L(\tau', \gg'_t)$ and $L(\tau'', \gg''_t)$ are subsets of the factor languages L'_{target} and L''_{target}, as stated in (22).

(22) *Step IV:*
$L(\tau', \gg'_t) \subseteq L'_{\text{target}}$ and $L(\tau'', \gg''_t) \subseteq L''_{\text{target}}$.

Finally, factorizability of the underlying typology plays a role again, as it ensures that the intersection of the two factor languages L'_{target} and L''_{target} coincides with the target language L_{target} the algorithm is trained on in the original run, as stated in (23).

(23) *Step V:*
$L_{\text{target}} = L'_{\text{target}} \cap L''_{\text{target}}$.

I can now put the pieces together as in (24), where (24a) holds by Step I, (24b) holds by Step IV, and (24c) holds by Step V.

(24) a. $L(\tau, \gg_t) = L(\tau', \gg'_t) \cap L(\tau'', \gg''_t)$
b. $\subseteq L'_{\text{target}} \cap L''_{\text{target}}$
c. $= L_{\text{target}}$

In the end, (24) provides the desired inclusion $L(\tau, \gg_t) \subseteq L_{\text{target}}$, that shows that the GLA is restrictive on the original typology. □

4 How to get the GLA to factorize

Theorem 1 motivates the following question: under which conditions can the GLA be guaranteed to factorize when run on a factorizable typology? This Section addresses this question, and shows that the proper definition (4) of the subroutine for the choice of the current loser form bears on the issue.

To appreciate the problem, consider the following slight variant (25) of the original factorizable iteration (12).

(25)

	current vector $\boldsymbol{\theta}_{t-1}$		updated vector $\boldsymbol{\theta}_t$
C_1=IDENT[ROUND]	20		**21**
C_2=*[+ROUND]	30		**29**
C_3=IDEND[HIGH]	17	(/y/, [y], [æ]) →	**18**
C_4=*[+HIGH]	15		**14**
C_5=IDEND[LOW]	5		**6**
C_6=*[−HIGH, −LOW]	3		3

Again, at step (1a) the algorithm is fed the piece of data (/y/, [y]) that pairs the underlying form /y/ with the faithful winner [y]. Yet, suppose that the algorithm now picks the loser form [æ] at step (1b), rather than the loser form [i] prescribed by Tesar and Smolensky's (1998) subroutine (4) for the choice of the current loser form. The three faithfulness constraints C_1 = IDENT[ROUND], C_3 = IDENT[HIGH] and C_5 = IDENT[LOW] are all winner-preferring relative to the underlying/winner/loser form triplet (/y/, [y], [~~æ~~]). The markedness constraints C_2 = *[+ROUND] and C_4 = *[+HIGH] are instead loser-preferring. As the loser-preferrer C_2 is ranked above all three winner preferrers C_1, C_3, C_5 according to the current ranking vector $\boldsymbol{\theta}_{t-1}$, update is triggered. The loser-preferrers C_2 and C_4 are demoted by 1 and the winner-preferrers C_1, C_3, and C_5 are promoted by 1, yielding the updated ranking vector $\boldsymbol{\theta}_t$.

Let $\boldsymbol{\phi}$ and $\boldsymbol{\varphi}$ denote the generic ranking vectors for the two factor constraint sets $\{C_1, C_2\}$ and $\{C_3, C_4, C_5, C_6\}$ in (7b) and (8b), respectively. The tth iteration in two possible runs of the GLA on the factor typologies (7) and (8) are provided in (26a) and (26b).

(26) a.

	current vector $\boldsymbol{\phi}_{t-1}$		updated vector $\boldsymbol{\phi}_t$
C_1=IDENT[ROUND]	20		**21**
C_2=*[+ROUND]	30	(/y/, [y], [æ]) →	**29**

b.

	current vector $\boldsymbol{\varphi}_{t-1}$		updated vector $\boldsymbol{\varphi}_t$
C_3=IDEND[HIGH]	17		17
C_4=*[+HIGH]	15	(/y/, [y], [æ]) →	15
C_5=IDEND[LOW]	5		5
C_6=*[−HIGH, −LOW]	3		3

Within the factor constraint subset $\{C_1, C_2\}$, the markedness constraint $C_2 =$ *[+ROUND] is loser-preferring and ranked above the winner-preferring $C_1 =$ IDENT[ROUND] according to the current ranking vector $\boldsymbol{\phi}_{t-1}$ in the factor run (26a). Thus update is performed, yielding the ranking vector $\boldsymbol{\phi}_t$. Within the factor constraint set $\{C_3, C_4, C_5, C_6\}$, the markedness constraint $C_4 =$ *[+HIGH] is loser-preferring, but it is already ranked underneath the winner-preferring constraint $C_3 =$ IDEND[HIGH] according to the current ranking vector $\boldsymbol{\varphi}_{t-1}$, so that no update is performed in the factor run (26b).

The current ranking vector $\boldsymbol{\theta}_{t-1}$ in (25) can be assembled from the two current ranking vectors $\boldsymbol{\phi}_{t-1}$ and $\boldsymbol{\varphi}_{t-1}$ in (26): the ranking values assigned by $\boldsymbol{\theta}_{t-1}$ to constraints in $\{C_1, C_2\}$ and $\{C_3, C_4, C_5, C_6\}$ coincide with the ranking values assigned by $\boldsymbol{\phi}_{t-1}$ and $\boldsymbol{\varphi}_{t-1}$, respectively. Crucially, the update *does not* preserves this property. In fact, the updated ranking vector $\boldsymbol{\theta}_t$ in (25) assigns to the constraints in $\{C_3, C_4, C_5, C_6\}$ ranking values that are different from those assigned by $\boldsymbol{\varphi}_t$. In other words, the original iteration (25) does not factorize, namely it cannot be reconstructed from the two corresponding factor iterations. The crucial difference between the factorizable iteration (12) and the non factorizable iteration (25) has to do with the choice of the loser form. The following theorem says that setting the current loser equal to the current prediction as prescribed by Tesar and Smolensky's (1998) subroutine (4) ensures the GLA's factorizability. This result thus provides a justification of this subroutine from the perspective of factorizability.

Theorem 2 *The GLA factorizes when run on a factorizable OT typology provided that the subroutine for the choice of the loser form at step (1b) always picks the form predicted by (an arbitrary refinement of) the current ranking vector, as prescribed by (4).* ■

Proof Consider the original run (15) of the GLA on the original typology $\tau = (\mathcal{X}, \mathcal{Y}, Gen, \mathcal{C})$, whose current ranking vector is denoted by $\boldsymbol{\theta}_t$. Assume that the typology τ factorizes relative to a partition $\mathcal{C}', \mathcal{C}''$ of the constraint set $\mathcal{C}$. Consider the corresponding factor runs (16), whose current ranking vectors are denoted by $\boldsymbol{\phi}_t$ and $\boldsymbol{\varphi}_t$. I will prove the factorizability identities (17b) by induction on time t. Assume that they hold at the end of the $(t-1)$th iteration, namely that $\boldsymbol{\theta}'_{t-1} = \boldsymbol{\phi}_{t-1}$ and $\boldsymbol{\theta}''_{t-1} = \boldsymbol{\varphi}_{t-1}$. Assume furthermore that the underlying/winner/loser form triplet $\mathbf{d}_t = (/x/, [y], [\not{z}])$ considered at the tth iteration has the property that the loser $[z]$ is the winner form predicted for the underlying form $/x/$ by (an arbitrary refinement of) the current ranking vector $\boldsymbol{\theta}_{t-1}$, as prescribed by (4). I will then prove that the factorizability identities (17b) also hold at time t, namely that $\boldsymbol{\theta}'_t = \boldsymbol{\phi}_t$ and $\boldsymbol{\theta}''_t = \boldsymbol{\varphi}_t$.

To start, suppose that the current underlying/winner/loser form triplet $\mathbf{d}_t$ is consistent with the current ranking vector $\boldsymbol{\theta}_{t-1}$, so that no update is performed in the original run, and $\boldsymbol{\theta}_t$ therefore coincides with $\boldsymbol{\theta}_{t-1}$. In this case, it is sufficient to show that also the corresponding ranking vectors $\boldsymbol{\phi}_{t-1}$ and $\boldsymbol{\varphi}_{t-1}$ in the factor runs are consistent with the current triplet $\mathbf{d}_t$ so that no update is performed in the two factor runs either. To this end, consider two arbitrary refinements $\gg'_{t-1}$ and $\gg''_{t-1}$ of $\boldsymbol{\phi}'_{t-1}$ and $\boldsymbol{\varphi}''_{t-1}$, and let me show that

both refinements are consistent with $\mathbf{d}_t$. The hypothesis that the factorizability identities hold at time $t-1$ means that $\boldsymbol{\theta}'_{t-1} = \boldsymbol{\phi}_{t-1}$ and $\boldsymbol{\theta}''_{t-1} = \boldsymbol{\varphi}_{t-1}$. Thus in particular, $\gg'_{t-1}$ and $\gg''_{t-1}$ are also refinements of $\boldsymbol{\theta}'_{t-1}$ and $\boldsymbol{\theta}''_{t-1}$, respectively. This means in turn that $\gg'_{t-1}$ and $\gg''_{t-1}$ are the restriction to $\mathcal{C}'$ and $\mathcal{C}''$ of some ranking $\gg_{t-1}$ that is a refinement of $\boldsymbol{\theta}_{t-1}$. The hypothesis that $\boldsymbol{\theta}_{t-1}$ is consistent with the current triplet $\mathbf{d}_t$ entails in particular that this refinement $\gg_{t-1}$ is consistent with $\mathbf{d}_t$. Factorizability of the underlying typology then entails that also $\gg'_{t-1}$ and $\gg''_{t-1}$ are consistent with $\mathbf{d}_t$, as desired.

Suppose next that the current ranking vector $\boldsymbol{\theta}_{t-1}$ is not consistent with the current underlying/winner/loser form triplet $\mathbf{d}_t$, so that update is performed in the original run. I will now prove that $\boldsymbol{\theta}'_t = \boldsymbol{\phi}_t$; the other factorizability identity $\boldsymbol{\theta}''_t = \boldsymbol{\varphi}_t$ is proven analogously. If all constraints in $\mathcal{C}'$ are even relative to $\mathbf{d}_t$ (namely assign the same number of violations to the winner and the loser), then of course none of the constraints in $\mathcal{C}'$ will be updated neither in the original nor in the corresponding factor update, as the GLA only re-ranks winner- and loser-preferring constraints. In this case, the factorizability identity $\boldsymbol{\theta}'_{t-1} = \boldsymbol{\phi}_{t-1}$ at time $t-1$ immediately entails the desired factorizability identity $\boldsymbol{\theta}'_t = \boldsymbol{\phi}_t$ at time t.

Suppose now that there is a at least a constraint in $\mathcal{C}'$ which is either winner- or loser-preferring according to the current triplet $\mathbf{d}_t$. If $\mathcal{C}'$ contains no winner-preferring constraint, then it has got to contain at least a loser-preferring constraint, otherwise all the constraints in $\mathcal{C}'$ would be even. Since $\mathcal{C}'$ contains a loser-preferring constraint but no winner-preferring constraint, then no ranking vector on $\mathcal{C}'$ can ever be consistent with $\mathbf{d}_t$. This means in turn that the current ranking vector $\boldsymbol{\phi}_{t-1}$ in the first factor run is not consistent with the current triplet $\mathbf{d}_t$ and it will therefore be updated. As $\boldsymbol{\theta}'_{t-1}$ and $\boldsymbol{\phi}_{t-1}$ coincide, as they are both updated, and as they are updated in the same way (winner-preferrers go up and loser-preferrers go down by 1), then the updated ranking vectors $\boldsymbol{\theta}'_t$ and $\boldsymbol{\phi}_t$ coincide as well.

Finally, consider the remaining case, where $\mathcal{C}'$contains some constraint that is winner-preferring according to the current triplet $\mathbf{d}_t = (/x/, [y], \cancel{[z]})$. Recall that the current loser $[z]$ has been selected according to (4). This means that there is some ranking $\gg_{t-1}$ that is a refinement of the current ranking vector $\boldsymbol{\theta}_{t-1}$ and incorrectly maps $/x/$ to $[z]$. In particular, this ranking $\gg_{t-1}$ must therefore be consistent with the underlying/winner/loser form triplet $\overline{\mathbf{d}} = (/x/, [z], \cancel{[y]})$ obtained from $\mathbf{d}_t$ by swapping the winner $[y]$ with the loser $[z]$, as stated in (27).

(27) *Step I:*
The current ranking vector $\boldsymbol{\theta}_{t-1}$ admits a refinement $\gg_{t-1}$ that is consistent with the swapped underlying/winner/loser form triplet $\overline{\mathbf{d}}$.

As the winner form and the loser form in the two triplets $\mathbf{d}_t$ and $\overline{\mathbf{d}}$ have beed swapped, every constraint that is winner-preferring relative to $\mathbf{d}_t$ is loser-preferring relative to $\overline{\mathbf{d}}$ and vice versa. The hypothesis that $\mathcal{C}'$ contains some constraint that is winner-preferring relative to $\mathbf{d}_t$ thus entails that it contains some constraint that is loser-preferring relative to $\overline{\mathbf{d}}$, as stated in (28).

(28) *Step II:*
The constraint subset $\mathcal{C}'$ contains some constraint that is loser-preferring relative to the swapped triplet $\overline{\mathbf{d}}$.

Let $\gg'_{t-1}$ be the factor ranking induced by the ranking $\gg_{t-1}$ on the constraint subset $\mathcal{C}'$. Assume that the underlying typology factorizes. Thus, the fact (27) that the ranking $\gg_{t-1}$ is consistent with the swapped triplet $\overline{\mathbf{d}}$ entails that the factor ranking $\gg'_{t-1}$ is consistent with it as well, as stated in (29).

(29) *Step III:*
The current ranking vector $\boldsymbol{\theta}_{t-1}$ admits a refinement $\gg_{t-1}$ whose restriction $\gg'_{t-1}$ to $\mathcal{C}'$ is consistent with the swapped triplet $\overline{\mathbf{d}}$.

Since the factor ranking $\gg'_{t-1}$ on $\mathcal{C}'$ is consistent with $\overline{\mathbf{d}}$ by (29) and since furthermore $\mathcal{C}'$ contains a loser-preferring constraint relative to $\overline{\mathbf{d}}$ by (28), then it must be the case that $\mathcal{C}'$ contains some constraint $C \in \mathcal{C}'$ that is winner-preferring relative to $\overline{\mathbf{d}}$ and ranked by $\gg'_{t-1}$ above every constraint that is loser-preferring relative to $\overline{\mathbf{d}}$, as stated in (30).

(30) *Step IV:*
The current ranking vector $\boldsymbol{\theta}_{t-1}$ admits a refinement $\gg_{t-1}$ whose restriction $\gg'_{t-1}$ to $\mathcal{C}'$ has the following property: $\gg'_{t-1}$ ranks a constraint of $\mathcal{C}'$ that is winner-preferrer relative to $\overline{\mathbf{d}}$ above every constraint in $\mathcal{C}'$ that is instead loser-preferring relative to $\overline{\mathbf{d}}$.

Switching back from the swapped triplet $\overline{\mathbf{d}}$ to the original triplet $\mathbf{d}_t$, every winner- (respectively, loser-) preferring constraint becomes loser- (respectively, winner-) preferring. Fact (30) thus entails the following fact (31).

(31) *Step V:*
The current ranking vector $\boldsymbol{\theta}_{t-1}$ admits a refinement $\gg_{t-1}$ whose restriction $\gg'_{t-1}$ to $\mathcal{C}'$ has the following property: $\gg'_{t-1}$ ranks a constraint of $\mathcal{C}'$ that is loser-preferring relative to $\mathbf{d}_t$ above every constraint in $\mathcal{C}'$ that is instead winner-preferring relative to $\mathbf{d}_t$.

In other words, (31) says that the ranking sub-vector $\boldsymbol{\theta}'_{t-1}$ is not consistent with the current triplet $\mathbf{d}_t$. As $\boldsymbol{\phi}_{t-1}$ coincides with $\boldsymbol{\theta}'_{t-1}$ by the inductive hypothesis, then $\boldsymbol{\phi}_{t-1}$ is not consistent with $\mathbf{d}_t$ either, and thus an update is triggered at the tth iteration in the factor run as well. As $\boldsymbol{\theta}'_{t-1}$ and $\boldsymbol{\phi}_{t-1}$ coincide, as both are updated, and as they are updated in the same way, then the updated ranking vectors $\boldsymbol{\theta}'_t$ and $\boldsymbol{\phi}_t$ coincide as well. □

5 Conclusion

OT is a inherently comparative model of grammar: an underlying form is mapped to that surface form that wins the competition against all other loser forms. Assuming that the GLA is provided at each iteration with the desired mapping of an underlying form into the corresponding winner, the algorithm

needs to pick a corresponding loser in order to test its current grammar. Whenever possible, the GLA should pick an *informative* loser, namely one that is not already condemned by the current grammar, so that the GLA will be able to learn something from the current piece of data. Tesar and Smolensky suggest a specific implementation of this idea: the GLA should picks that specific informative loser that consists of the incorrect prediction made by the current grammar. Yet, the choice of this specific loser over any other informative loser requires some extra computation, and needs therefore some careful motivation. In this paper, I have provided one such motivation, from the perspective of factorizability. I have shown that Tesar and Smolensky's specific choice of the current loser allows a run of the GLA to factorize, whenever the underlying typology factorizes (Theorem 2). This means that, if the underlying typology encodes independent processes and can therefore be factorized into smaller independent typologies, then a run of the GLA on the original typology can be mimicked by corresponding runs on the factor typologies. And I have argued that factorizability is a desirable property of error-driven ranking algorithms such as the GLA, as it allows the analysis of the algorithm on the original large typology to be reduced to the simpler analyses on the smaller factor typologies (Theorem 1).

Acknowledgements I wish to thank Adam Albright for lots of help and discussion. I also wish to thank Paul Boersma, Alan Prince, Paul Smolensky, Donca Steriade, and Bruce Tesar, as well as an anonymous JLLI reviewer. This work has been supported in part by a grant from the *Fyssen Research Foundation* ("Child acquisition of sound patterns: a computational modeling approach").

References

Boersma P (1997) How we learn variation, optionality and probability. In: van Son R (ed) Proceedings of the Institute of Phonetic Sciences (IFA) 21, Institute for Phonetic Sciences, University of Amsterdam, pp 43–58

Boersma P (1998) Functional phonology. PhD thesis, University of Amsterdam, The Netherlands, the Hague: Holland Academic Graphics

Boersma P (2009) Some correct error-driven versions of the constraint demotion algorithm. Linguistic Inquiry 40:667–686

Boersma P, Hayes B (2001) Empirical tests for the Gradual Learning Algorithm. Linguistic Inquiry 32:45–86

Boersma P, Levelt C (2000) Gradual constraint-ranking learning algorithm predicts acquisition order. In: Clark EV (ed) Proceedings of the 30th Child Language Research Forum, CSLI, Stanford University, pp 229–237, corrected version available as ROA 361

Curtin S, Zuraw K (2002) Explaining constraint demotion in a developing system. In: Skarabela B, Fish S, Do AHJ (eds) Boston University Conference on Language Development (BUCLD) 26, Cascadilla Press, vol 1, pp 118–129

Kager R (1999) Optimality Theory. Cambridge University Press, Cambridge, United Kingdom

Magri G (2012) Robust analysis of error-driven ranking algorithms and its implications for modeling the child acquisition of phonotactics, accepted subject to revisions at the *Journal of Logic and Computation*

Prince A (2002) Entailed ranking arguments, ms., Rutgers University. Also available as ROA 500

Prince A, Smolensky P (2004) Optimality Theory: Constraint Interaction in Generative Grammar. Blackwell, as Technical Report CU-CS-696-93, Department of Computer Science, University of Colorado at Boulder, and Technical Report TR-2, Rutgers Center for Cognitive Science, Rutgers University, New Brunswick, NJ, April 1993. Also available as ROA 537 version

Tesar B, Smolensky P (1998) Learnability in Optimality Theory. Linguistic Inquiry 29:229–268