This is Chapter 4/5 only. Refer to 20240818 dissertation outline and notes for complete outline.

Kaili's very official dissertation draft

For purposes of graduating in the near-ish future

4/5. Learning biases and optimal learning conditions

As per usual, my initial work will be done in Google Docs while the final draft(s) will be in LaTeX. So any formatting issues or nuances will be sorted out later.

Yellow highlighting indicates updates/follow-up needed.

Notes and/or specific requests for feedback highlighted in blue throughout.

Prior to this chapter, I will have introduced:

- The balto-finnic languages / vowel pattern typology
- Representative sample languages (patterns) for the above
- My constraint set & OT analysis of the sample languages
- Learning algorithms, in particular GLA-type learners
    - Definition of idempotence
    - Definition of θ-notation for ranking values
    - Explain my method of calculating "average frequency of correct results"
- My methods for simulations - python script etc
- Simulated learning data for the sample languages

Simulating acquisition of a grammar with an algorithmic learner involves many potential variables, parameters, and biases. In this chapter I introduce those factors that are relevant to learning Balto-Finnic vowel patterns, and discuss the impact that each has on the sample languages Finnish, North Estonian, and North Seto.

In Section 4.1 I introduce the factors that are assumed to remain constant, present learning results given these foundational assumptions, and discuss challenges to be overcome from this starting point. Although there are a number of different obstacles that contribute to the difficulty of learning these languages in the context of their typology, they are not all immediately obvious. Therefore each of Sections 4.2, 4.3, and 4.4 introduces a particular learning bias that addresses an already-identified problem, while simultaneously uncovering more subtle obstacles not previously apparent. Section 4.2 investigates the specific-over-general faithfulness bias, which facilitates privileging first-syllable vowels but reveals that faithfulness constraints are nevertheless promoted too high. Section 4.3 focuses on options for varying the promotion rate applied at each learning update, which tempers the dramatic climb of the faithfulness constraints but shows that overly-specific markedness generalizations are being learned in some cases. Section 4.4 explores the general-over-specific markedness bias, which prioritizes more restrictive (more general) markedness constraints over less restrictive ones. Finally, in Section 4.5, I summarize all of the learning simulations performed with various combinations of values for the biases introduced in the preceding sections and generalize a set of ideal conditions for learning these Balto-Finnic languages.

4.1. Learning simulations with default settings

The GLA, as specified by Boersma and Hayes (2001) and discussed in Section 3.X, describes the general procedure for this type of gradual, error-driven learning. The bare bones of the learning algorithm as described lay the foundation for additional potential parameters or biases to be included.

4.1.1. Learning parameters/biases assumed to remain constant

For the purposes of this project, there are a number of parameters that I considered allowing to vary, but ultimately decided to keep constant.

The first of these determines whether all constraints have the same initial ranking values or if faithfulness constraints should start lower than markedness constraints. I consistently apply a low-faithfulness bias in these simulations.[1] The bias toward low initial faithfulness is widely used in the learning literature, as it helps to ensure that the acquired grammar is as restrictive as possible; that is, it mitigates the Subset Problem (Angluin 1980, Baker 1979). Readers can find more detailed discussion in, e.g., Gnanadesikan (1995), Smolensky (1996), Hayes (2004),

---

[1] I also ran a small number of exploratory simulations in which faithfulness constraints experience a more persistent downward bias, being demoted at regular intervals through the learning process. However, these experiments did not produce any promising results so I set the notion of “gravity” aside and did not pursue it any further.

Prince and Tesar (2004), and Jesney and Tessier (2011). The default implementation of this bias in this project is to set the initial ranking value of faithfulness constraints to be 0, and that of markedness constraints to be 100. There are some other biases discussed in later parts of this chapter that will set initial markedness values to be different from the default; however, these will continue to preserve the overarching low-faithfulness bias.

The second parameter that will remain constant is that of demotion eligibility; that is, whether all loser-preferring constraints get demoted at each learning update, or just the undominated ones. In all learning simulations, I demote *all* such constraints rather than choosing to run some simulations in which only *undominated* loser-preferrers get demoted. Boersma and Hayes (2001) find that demoting only undominated losers caused the GLA to fail on their test data. On the other hand, Magri (2012) shows that doing so can prevent the learner from converging efficiently. Suffice to say that even if choosing to demote all loser-preferrers affects the learner's ability to converge *efficiently*, it will not affect whether or not the learner converges *at all*.

The fourth constant is the number of learning trials, which is fixed at 20,000 for each simulation. All simulations described herein converged well before iterating through this many trials, providing a long enough timeline to ensure that even the odd later error (caused by a particularly noisy evaluation) did not affect the overall ranking.

The fifth parameter that will remain constant is permitting constraints to take on negative rankling values. Since I am working with ranked (classic OT) rather than weighted (e.g. Harmonic Grammar) constraints, there is no particular concern associated with negative ranking values; all ranking values are converted to relative ordinal rankings at evaluation so the actual numerical values themselves are irrelevant. For example, the values {θ_{C1} = 100, θ_{C2} = 50} produce the exact same ranking as the values {θ_{C1} = -25, θ_{C2} = -75}. Given this fact, the default OTSoft (Hayes et al, 2013a) approach for GLA learning is used; that is, to permit demotion of constraints even when the resulting ranking value is negative. [I was sure that I'd seen someone else make the argument that negative values are ok when using GLA with classic OT evaluation, but I can't for the life of me find it. Does this ring a bell for anyone else? I'm happy to explain it myself but it seems silly to do so if it already exists elsewhere!]

The last few parameters that are held constant across simulations are the organization of learning trials into stages, evaluation noise, and the plasticity function. None of the results I discuss appear to depend on changes to these settings, so the default OTSoft (Hayes et al, 2013a) assumptions for GLA learning are used; they are summarized in Table 1.

Parameter	**Stage 1**	**Stage 2**	**Stage 3**	**Stage 4**
Number of learning trials	5000	5000	5000	5000
Evaluation noise	2	2	2	2
Plasticity	2	0.2	0.02	0.002

Table 1. Invariant settings for GLA learning.

4.1.2. Simulation results

Initial learning simulations use all of the basic parameters (those described in Section 4.1.1) at their default settings, with no additional biases or parameters introduced. Learner A is defined with the settings in Table 2.

Parameter	Setting
All basic parameters	Default

Table 2. Parameter settings for Learner A.

Under these default conditions, learning simulations for all three sample languages fail to acquire the target grammars, producing instead fully-faithful grammars. Such grammars succeed 100% of the time on the input forms, since we assume idempotence. However, they are very poorly equipped to deal with illicit test forms. Test results are summarized in Table 3.

Language	Average frequency of correct outputs
Finnish	0.2654
North Estonian	0.2455
North Seto	0.2982
Overall	0.2697

Table 3. Summary of results from simulations with Learner A.

Finnish: Table 4 shows the final ranking values for a selection of crucial constraints, after learning from simulated Finnish data. Id(Bk) has risen to the top of the rankings, producing a grammar that does not align with the crucial rankings proposed in Section 3.X:

$*B_2$, $*F_3…\underline{B}_5$, $*F_3\underline{B}_5$, $*\underline{B}_5…F_3$, $*\underline{B}_5F_3$ >> Id(Bk)σ_1 >> Id(Bk)

Constraint	Final ranking value
Id(Bk)	116
$*B_2$	110
$*F_3…\underline{B}_5$	106
$*\underline{B}_5…F_3$	106
…	…
$*F_3\underline{B}_5$	104
$*\underline{B}_5F_3$	104
…	…
Id(Bk)σ_1	80

Table 4. Excerpt of final ranking values for Finnish after simulation with Learner A.

North Estonian: The final ranking values for a selection of crucial constraints, after learning from simulated North Estonian inputs, are shown in Table 5. Id(Bk) is at the top of this grammar too, meaning that it does not achieve the crucial rankings proposed in Section 3.X:

$*B_1 >> Id(Bk)\sigma_1 >> *F_3, *B_2 >> Id(Bk)$

Constraint	Final ranking value
Id(Bk)	116
$*B_1$	104
$*\underline{B}_5…F_3$	104
$*\underline{B}_5F_3$	104
…	…
$*F_3$	102
$*B_2$	102
…	…
$*F_5…\underline{B}_2$	100
$*F_5\underline{B}_2$	100
…	…
$Id(Bk)\sigma_1$	92
…	…

Table 5. Excerpt of final ranking values for North Estonian after simulation with Learner A. A selection of no-disagreement constraints are included here for the purpose of comparison with results of simulations discussed in subsequent sections.

North Seto: Learning from simulated North Seto data results in final ranking values for a selection of crucial constraints shown in Table 6. Id(Bk) has once again risen to the top; the resulting grammar does not meet the crucial target rankings proposed in Section 3.X:

$*F_4…\underline{B}_5, *F_4\underline{B}_5, *\underline{B}_5…F_4, *\underline{B}_5F_4, Id(Bk)syl1 >> *B1 >> Id(Bk)$

Constraint	Final ranking value
Id(Bk)	116
$*F_4…\underline{B}_5$	106
$*\underline{B}_5…F_4$	106
$*F_4\underline{B}_5$	106
…	…
$*B_1$	102
…	…
$*\underline{B}_5F_4$	100
…	…
$Id(Bk)\sigma_1$	80

Table 6. Excerpt of final ranking values for North Seto after simulation with Learner A.

4.1.3. Discussion

There are several obstacles that must be addressed on the way to acquiring better – even excellent – final grammars. However, in the results shown above in Section 4.1.2, not all of the challenges are apparent; some only become clear as the initial problems are resolved. In this section I discuss those that are immediately identifiable, and leave the others to be discussed and addressed in subsequent sections of this chapter.

With respect to the results presented in Section 4.1.2, the most glaring problem is that Id(Bk) is highest ranked in all three. This means that during the learning process, Id(Bk) rises all the way from its initial value of 0, past all of the markedness constraints starting at 100, to the very top of the rankings. Such grammars are fully faithful and therefore overgenerate to the point of excluding nothing.

The reason for Id(Bk)'s rise all the way to the top of the rankings is due to a lack of *a priori* relative ranking of the specific and general faithfulness constraints (specifically, an obligation for Id(Bk)σ_1 to outrank Id(Bk)) under assumption of idempotence. Each time the learner encounters an error, Id(Bk) is always a winner-preferring constraint, since the underlying form is assumed to be identical to the heard surface form. The ERC matrix in Table 7 shows that for a surface form of /o..ɑ/ (grammatical in all three of the sample languages), Id(Bk) is a winner-preferrer for *any* error and Id(Bk)σ_1 is a winner-preferrer for only those errors involving the first syllable. Thus an error in the first syllable will result in promotion of both faithfulness constraints, but any error past the first syllable will result in promotion of only the general one. Since a crucial element of all three sample languages' target grammars is for Id(Bk)σ_1 to outrank Id(Bk), this ranking will never be achieved and the learner will only stop making errors once Id(Bk) has been promoted all the way to the top of the rankings.

/o..ɑ/	**markedness constraints**	**ID(Bk)**	**ID(Bk)-σ_1**
o..ɑ ~ o..æ	…	W	
o..ɑ ~ ø..ɑ	…	W	W
o..ɑ ~ ø..æ	…	W	W

Table 7. ERC matrix demonstrating that under assumption of idempotence, all learning errors have Id(Bk) as a winner-preferring constraint.

Addressing the relative ranking of specific vs general faithfulness constraints is not the only obstacle to successful learning of grammars for the sample languages. However, as it is the only one apparent under the learning conditions presented in Section 4.1, it must be addressed before any others can be revealed. Section 4.2 presents a solution for this problem.

4.2. F_{spec} >> F_{gen}

The constraint set that I use for this project includes only two faithfulness constraints, **ID(Bk)** and **ID(Bk)-σ_1**, the first applying more broadly and the second in a narrower context. When two such versions of a faithfulness constraint exist, it is possible to construct a grammar in which marked elements in underlying forms surface only in privileged contexts. For example, the ranking **ID(Bk)-σ_1** >> ***F_3** >> **ID(Bk)** bans vowels in set **F_3** in general, but permits them in initial syllables.

A specific-over-general faithfulness bias (F_{spec} >> F_{gen}) is a strategy that can help find the most restrictive grammar that accounts for the input data, avoiding a superset (overgenerating) grammar (Hayes 2004, Tessier 2007). I take two slightly different approaches to this idea, detailed in Sections 4.2.1 and 4.2.2.

4.2.1. *A priori* bias

One approach to the specific-over-general faithfulness bias is to ensure that the ranking value for the specific version of the constraint is a minimum specified distance higher than that of the general version [TODO citation?]. The satisfaction of this bias is checked persistently through the learning simulation, both in the initial state and after each individual learning update.

4.2.1.1. Rationale

Maintaining a minimum difference between the ranking values of a specific-general pair of faithfulness constraints ensures that the specific version of the constraint always has a better opportunity to claim credit for a particular output form than the general one does, corresponding to a more restrictive grammar overall.

4.2.1.2. Implementation

The F_{spec} >> F_{gen} bias between any specific-general pair of faithfulness constraints can be implemented by means of an *a priori* bias that ensures $\theta_{Fspec} - \theta_{Fgen} \geq d$, for some distance *d*. Practically, the learner adjusts the initial ranking values such that any two constraints in this type of relationship are at least *d* apart, and then does the same after each learning update. If the two constraints have a difference of less than *d*, then it is always the case that the specific one has its value increased rather than the general one having its value decreased. OTSoft (Hayes et al, 2013a) sets the default value of this difference to be *d = 20*, stating that it is “very close probabilistically to being an obligatory ranking” (Hayes et al, 2013b: 24).

In my learning simulations, I test the omission of this bias as well as a range of different *d* values: 0 (θ_{Fspec} must be no less than θ_{Fgen}), 10, 20, 30, and 40.

4.2.1.3. Simulation results - *a priori*

To demonstrate the effect of the *a priori* bias, I simulate acquisition of the three sample languages using Learner B, defined with the settings in Table 8. The selection of *d* = 20 for illustrative purposes is drawn from the OTSoft default as mentioned above. Results using learners with other values of *d* are summarized in Section 4.5.

Parameter	Setting
All basic parameters	Default
A priori bias	d = 20

Table 8. Parameter settings for Learner B.

With the *a priori* bias set to d = 20, learning simulations for all three sample languages fail to acquire the target grammars. The learner trained on North Estonian data produces a grammar that, while not correct, does have some promising characteristics. On the other hand, the learners trained on Finnish and North Seto data once again produce fully-faithful grammars. Test results are summarized in Table 9.

Language	Average frequency of correct outputs
Finnish	0.2895
North Estonian	0.7796
North Seto	0.3133
Overall	0.4608

Table 9. Summary of results from simulations with Learner B.

Finnish: Table 10 shows the final ranking values for a selection of crucial constraints, after learning from simulated Finnish data. Both faithfulness constraints have risen to the top. Id(Bk)'s distance above $*B_2$ and the relevant no-disagreement constraints is small enough that evaluation noise might cause it to swap rankings with one of its neighbours. However, in order to meet the crucial rankings

$*B_2$, $*F_3\ldots\underline{B}_5$, $*F_3\underline{B}_5$, $*\underline{B}_5\ldots F_3$, $*\underline{B}_5F_3$ >> Id(Bk)σ_1 >> Id(Bk)

proposed in Section 3.X, such swaps would have to be guaranteed to occur at every evaluation, which is extremely unlikely given the final ranking values. Hence the final grammar produced by Learner B on Finnish inputs is more or less fully faithful, with a representative evaluation shown in Tableau (1).

Constraint	Final ranking value
Id(Bk)σ_1 Id(Bk) $*\underline{B}_5\ldots F_3$ $*B_2$ $*F_3\underline{B}_5$ $*F_3\ldots\underline{B}_5$ $*\underline{B}_5F_3$	136.000 116.000 112.000 110.000 110.000 110.000 110.000

…	…

Table 10. Excerpt of final ranking values for Finnish after simulation with Learner B.

(1) Sample evaluation of input /o..æ/ in the Finnish grammar acquired by Learner B. The grammar selects the faithful candidate [o..æ] as optimal even though it is not harmonic.

	/o..æ/	Id(Bk)σ_1	Id(Bk)	*$\underline{B}_5\ldots F_3$	… (constraints such as *B_2, *$F_3\underline{B}_5$, *$F_3\ldots\underline{B}_5$, *$\underline{B}_5F_3$, and others)
☞	o..æ			*	
	o..ɑ		*!		
	ø..æ	*!	*		
	ø..ɑ	*!	**		

[TODO possibly adjust the following paragraph / table after 20241128 discussion with AMT]

In theory it should have been reasonable for Id(Bk)σ_1 to end up with a final ranking value greater than or equal to the top-ranked markedness constraints with Id(Bk) lower down. However, at the time that Id(Bk)σ_1 approaches the highest-ranked markedness constraints (including *B_2 with θ=110), the other context-free markedness constraints all have values in [100, 106] and are therefore within a small enough window for evaluation noise to make (e.g.) *B_3 or *F_3 active in selecting the optimal candidate (see Table 10.1). This results in errors and therefore more updates which push the faithfulness constraints ever higher. It is only once Id(Bk) surpasses this clump of markedness constraints that errors taper off and the learner converges.

Trial number 69 i..e..ɑ ~ ɯ..ɤ..ɑ	Trial number 73 u..u..ɑ ~ u..y..æ	Trial number 90 u..y..ø ~ i..u..o
*B_2 110	Id(Bk)σ_1 112	Id(Bk)σ_1 120
Id(Bk)σ_1 108	*B_2 110	*B_2 110
*B_3 106	*B_3 108	*$\underline{B}_5\ldots F_3$ 108
*F_3 104	*$\underline{B}_5\ldots F_3$ 106	*B_3 106
*B_1 104	*F_3 104	*$F_3\underline{B}_5$ 106
*$\underline{B}_5\ldots F_3$ 104	*B_1 104	*$F_3\ldots\underline{B}_5$ 106
*F_1 102	*$F_3\ldots\underline{B}_3$ 104	*$\underline{B}_5F_3$ 106
*$F_3\underline{B}_3$ 102	*$F_3\underline{B}_5$ 104	*F_3 104
*$F_3\ldots\underline{B}_3$ 102	*$F_3\ldots\underline{B}_5$ 104	*B_1 104
*$F_3\underline{B}_5$ 102	*$\underline{B}_5F_3$ 104	*$F_3\ldots\underline{B}_3$ 104
*$F_3\ldots\underline{B}_5$ 102	*$F_3\underline{B}_3$ 102	*$\underline{B}_5F_1$ 104
*$\underline{B}_3\ldots F_3$ 102	*$F_4\ldots\underline{B}_3$ 102	*$\underline{B}_5\ldots F_1$ 104
*$\underline{B}_5F_1$ 102	*$\underline{B}_3\ldots F_3$ 102	*F_1 102

$*\underline{B}_5 \ldots F_1$ 102	$*\underline{B}_5 F_1$ 102	$*F_3 \underline{B}_3$ 102
$*\underline{B}_5 F_3$ 102	$*\underline{B}_5 \ldots F_1$ 102	$*\underline{B}_3 \ldots F_3$ 102
$*F_4$ 100	$*F_1$ 100	Id(Bk) 100
$*F_5$ 100	$*F_4$ 100	$*F_4$ 100
$*B_5$ 100	$*F_5$ 100	$*F_5$ 100
…	$*B_5$ 100	$*B_5$ 100
Id(Bk) 88	…	…
	Id(Bk) 92	

Table 10.1. The highest of Finnish Learner B's constraint ranking values after three different learning updates. Although the crucial constraints (ideally $*B_2$ >> Id(Bk)σ_1 >> $*\underline{B}_5 \ldots F_3$, $*F_3\underline{B}_5$, $*F_3 \ldots \underline{B}_5$, $*\underline{B}_5 F_3$) are in reasonably good positions, constraints such as $*B_3$, $*B_5$, $*F_1$, $*F_3$, $*F_4$, $*F_5$ are near enough to be potentially disruptive.

North Estonian: The final ranking values for a selection of crucial constraints, after learning from simulated North Estonian inputs, are shown in Table 11. Several of the crucial relative rankings $*B_1$ >> Id(Bk)σ_1 >> $*F_3$, $*B_2$ >> Id(Bk), proposed in Section 3.X, are met by this grammar. However, one of the key elements – the full ban on vowels from set B_1 – is missing, by virtue of that fact that $*B_1$'s final value is not only not at the top, but below even Id(Bk). Thus the acquired grammar will incorrectly permit B_1 vowels in initial syllables.

Constraint	**Final ranking value**
Id(Bk)σ_1	129.220
$*B_2$	115.000
…	…
$*F_3$	110.220
Id(Bk)	109.220
$*B_1$	108.000
$*\underline{B}_5 \ldots F_3$	108.000
…	…
$*F_5 \ldots \underline{B}_2$	106.000
…	…
$*F_5 \underline{B}_2$	104.600
…	…
$*\underline{B}_5 F_3$	104.000
…	…

Table 11. Excerpt of final ranking values for North Estonian after simulation with Learner B. A selection of no-disagreement constraints are included here for the purpose of comparison with results of simulations discussed in subsequent sections.

The ranking acquired by this learner does generally follow the required positional restrictions by ranking Id(Bk)σ_1 >> $*F_3$, $*B_2$ >> Id(Bk); however, the ranking values are close enough together that the stochastic nature of evaluation results in somewhat variable adherence to these positional restrictions. For example, ungrammatical test input /y..æ/ would be expected to surface as [y..ɑ], neutralizing the restricted vowel in the second syllable. However, during testing, this grammar selects output candidates with the frequencies shown in Table 12.

/y..æ/	Output frequency
y..æ	0.35
y..ɑ	0.65
u..æ	0.00
u..ɑ	0.00

Table 12. Frequency of candidate selection for input /y..æ/ with North Estonian grammar acquired by Learner B. Number of sample evaluations = 100.

Although there is some crowding, at least one success the North Estonian learner achieves that the Finnish one does not is that the learner converges with several markedness constraints between the two faithfulness constraints. [TODO add some explanation here after 20241128 discussion with AMT]

North Seto: Table 13 shows the final ranking values for a selection of crucial constraints, after learning from simulated North Seto data. Similar to the Finnish results, both faithfulness constraints have risen to the top. Though the markedness constraints are correctly ordered relative to each other, the relative positions of the faithfulness vs the markedness constraints are not correct with respect to the crucial rankings proposed in Section 3.X:

$*F_4\ldots\underline{B}_5$, $*F_4\underline{B}_5$, $*\underline{B}_5\ldots F_4$, $*\underline{B}_5F_4$, Id(Bk)syl1 >> *B1 >> Id(Bk)

Again, the final grammar produced by Learner B on North Seto inputs is essentially fully faithful, with similar learning challenges as described for Finnish.

Constraint	Final ranking value
Id(Bk)σ_1 Id(Bk) $*F_4\ldots\underline{B}_5$ $*\underline{B}_5\ldots F_4$ $*F_4\underline{B}_5$ $*\underline{B}_5F_4$ … $*B_1$ …	136.000 116.000 110.000 110.000 108.000 108.000 … 104.000 …

Table 13. Excerpt of final ranking values for North Seto after simulation with Learner B.

Results from both Section 4.2.1 and Section 4.2.2 are discussed in Section 4.2.4.

4.2.2. Favour specificity

The underlying idea for the favour-specificity bias is to allow the specific faithfulness constraint to rise independently of the general one, similar to the *Favour Specificity* principle that Hayes (2004) introduces for the Low-Faithfulness Constraint Demotion algorithm. Although that proposal focuses on a different algorithm, the same principle can be adapted to apply to the GLA as well.

4.2.2.1. Rationale

As discussed in 4.2.1, setting an *a priori* bias helps specific faithfulness constraints stay above their general counterparts. However, because each violation of a first-syllable faithfulness constraint is also necessarily a violation of a general faithfulness constraint, there is no opportunity for the specific constraint to ever rise any further above the general version than the *a priori* bias specifies. That is, it is always the case that either the pair of constraints is moving in tandem (if there is an error in the first syllable only) or the general constraint is "pushing" the specific one up form below (if there is at least one error in a non-initial syllable). Both of these scenarios have the same effect: the specific constraint does not ever move independently of the general one. Recall the ERC matrix in Table 7 for an illustration of this phenomenon.

This type of movement, where specific and general constraints are separated by what is effectively a constant distance, can cause a challenge for the learner in that the *d* value that is specified for the *a priori* bias may or may not be large enough for other necessary constraints and/or interactions to "fit" between the two faithfulness constraints, depending on the target grammar. For instance, suppose the target grammar has crucial rankings M_1 >> F_1 >> F_2 >> M_2, and the learner is set to its task with a fixed value (e.g., d = 20) assigned to the *a priori* bias. The F_1 >> F_2 relationship will be effectively categorical, which is sufficient for this grammar. However, suppose the target grammar has instead crucial rankings F_1 >> M_1 >> F_2 >> M_2. In this case, d = 20 does not create enough space: the constraints in either of the crucial rankings F_1 >> M_1 or M_1 >> F_2 (or both) will have ranking values close enough that evaluation noise will create some variability in surface forms. Conversely, attempting to solve this problem by arbitrarily setting the a priori bias to be larger can cause other issues instead (for example, it would prevent the learning of a target grammar where F_1 >> M_1 >> F_2 but M_1 must be variably interchangeable with both F_1 and F_2).

To address this challenge, I take an approach that allows the space between specific and general counterparts to change, depending on the kinds of errors that are made.

4.2.2.2. Implementation

As always, when a learning error triggers an update to the constraint ranking values, the relevant ERC is inspected for winner-preferring vs loser-preferring constraints. In this case, if both the specific and the general version of a particular faithfulness constraint are eligible for promotion (i.e., both prefer the winner), then only the specific one gets promoted.[2]

There is also a set of optional variations to this implementation, in which the *a priori* bias (if any; see Section 4.2.1) increases if the current θ_{Fspec} - θ_{Fgen} difference is greater than d. This increase is applied after all constraints involved in a learning update have been adjusted, and sets the new $d' = d + \Delta$. The three variations are (*only* occurring if $\theta_{Fspec} - \theta_{Fgen} > d$):

- Variation 1: $\Delta = (\theta_{Fspec} - \theta_{Fgen}) - d$ — expand the entire current difference
- Variation 2: $\Delta = ((\theta_{Fspec} - \theta_{Fgen}) - d)/2$ — expand half of the current difference
- Variation 3: $\Delta = ((\theta_{Fspec} - \theta_{Fgen}) - d)/i$, where i is the current learning trial number — expand a decreasing fraction of the current difference

[Have I done a poor job of explaining the variations? That last paragraph feels sort of convoluted. Also I'm not sure how far I want to actually go on this front, since it was not useful.]

4.2.2.3. Simulation results - Favour Specificity

To demonstrate the effect of the Favour Specificity bias, I simulate acquisition of the three sample languages using Learner C, defined with the settings in Table 14.

Parameter	Setting
All basic parameters	Default
Favour Specificity bias	Active

Table 14. Parameter settings for Learner C.

With the Favour Specificity bias applied, learning simulations for all three sample languages fail to acquire the target grammars. Once again, the grammar acquired by the learner trained on North Estonian is a significant improvement over the one acquired by Learner A, but the Finnish and North Seto grammars are essentially full faithful. Test results are summarized in Table 15.

Language	Average frequency of correct outputs
Finnish	0.2931
North Estonian	0.8998
North Seto	0.4260
Overall	0.5397

Table 15. Summary of results from simulations with Learner C.

[2] The general one still counts toward W, the total number of winner-preferring constraints, but it will not be affected by the update.

Based solely on its average frequency of correct outputs, North Seto Learner C appears to have shown some improvement over Learner A. However, this is in fact a statistically convenient side effect of final ranking values that are no better from a theoretical perspective; further explanation is provided below.

Finnish: Table 16 shows the final ranking values for a selection of crucial constraints, after learning from simulated Finnish data. Once again, both faithfulness constraints have risen to the top and the final grammar does not meet the crucial rankings proposed in Section 3.X:

$*B_2$, $*F_3\ldots\underline{B}_5$, $*F_3\underline{B}_5$, $*\underline{B}_5\ldots F_3$, $*\underline{B}_5F_3$ >> $Id(Bk)\sigma_1$ >> $Id(Bk)$

Rather, the final grammar produced by Learner B on Finnish inputs is more or less fully faithful.

Constraint	**Final ranking value**
$Id(Bk)\sigma_1$	116.000
$Id(Bk)$	112.000
$*F_3\ldots\underline{B}_5$	110.000
$*\underline{B}_5\ldots F_3$	110.000
$*B_2$	108.000
$*\underline{B}_5F_3$	108.000
…	…
$*F_3\underline{B}_5$	104.000
…	…

Table 16. Excerpt of final ranking values for Finnish after simulation with Learner C.

North Estonian: The final ranking values for a selection of crucial constraints, after learning from simulated North Estonian inputs, are shown in Table 17. As for Learner A, several of the crucial relative rankings $*B_1$ >> $Id(Bk)\sigma_1$ >> $*F_3$, $*B_2$ >> $Id(Bk)$, proposed in Section 3.X, are met by this grammar. The issue of the full ban is still relevant - $*B_1$ is still not at the top of the rankings - but at least it is above Id(Bk).

Constraint	**Final ranking value**
$Id(Bk)\sigma_1$	124.000
$*F_3$	112.002
$*B_2$	112.000
$*\underline{B}_5\ldots F_3$	110.000
…	…
$*B_1$	106.000
$*F_5\ldots\underline{B}_2$	106.000
$*\underline{B}_5F_3$	106.000
…	…
$*F_5\underline{B}_2$	104.000
…	…
Id(Bk)	70.002
…	…

Table 17. Excerpt of final ranking values for North Estonian after simulation with Learner C. A selection of no-disagreement constraints are included here for the purpose of comparison with results of simulations discussed in subsequent sections.

There is also a great deal more space between Id(Bk)σ_1 and Id(Bk), allowing for more-categorical relationships between the constraints of interest. For example, when given ungrammatical test input /y..æ/, Table 18 shows that this grammar selects the intended output [y..ɑ] in 100% of test evaluations (compare with only 65% for Learner B as shown in Table 12).

/y..æ/	Output frequency
y..æ	0.00
y..ɑ	1.00
u..æ	0.00
u..ɑ	0.00

Table 18. Frequency of candidate selection for input /y..æ/ with North Estonian grammar acquired by Learner C. Number of sample evaluations = 100.

North Seto: Table 19 shows the final ranking values for a selection of crucial constraints, after learning from simulated North Seto data. Similar to the Finnish results, both faithfulness constraints have risen to the top. As for Learner B, though the markedness constraints are correctly ordered relative to each other, the relative positions of the faithfulness vs the markedness constraints are not correct with respect to the crucial rankings proposed in Section 3.X:

$*F_4\ldots\underline{B}_5$, $*F_4\underline{B}_5$, $*\underline{B}_5\ldots F_4$, $*\underline{B}_5F_4$, Id(Bk)syl1 >> *B1 >> Id(Bk)

Constraint	Final ranking value
Id(Bk)σ_1	118.000
Id(Bk)	114.220
$*F_4\ldots\underline{B}_5$	112.000
$*\underline{B}_5\ldots F_4$	112.000
$*F_4\underline{B}_5$	110.000
$*\underline{B}_5F_4$	110.000
…	…
$*B_1$	108.020
…	…

Table 19. Excerpt of final ranking values for North Seto after simulation with Learner C.

As noted earlier in this section (with reference to Table 15), the average frequency of correct results is higher than for the grammar acquired by Learner B, even though the final ranking values show a constraint ordering that appears to be fully faithful. This difference is due to the spacing between constraints– in particular, Id(Bk) vs $*B_1$ and the relevant VH constraints. In the final grammar acquired by Learner C, all of the constraints are much closer together and therefore the stochastic evaluation is more likely to result in Id(Bk) swapping places with one or more of the markedness constraints, generating outputs that are more likely to obey markedness (vowel harmony and/or positional restrictions) rather than faithfulness pressures.

4.2.3. Simulation results - *a priori* and Favour Specificity

[Not sure if I should include this section or not, with results from a learner that has both of the specific-faithfulness biases applied. The results look pretty similar to what we get from the learners with either of the biases above, depending on the language. Is it useful to talk about *why* that's the case? I don't really think so… On the other hand, I would like to use both of them combined in the later sections of this chapter, since the combination often seems to produce better results (when also combined with whatever other bias) than just one or the other. Wondering if it's appropriate to include both of the spec-F biases as a base for those more complex simulations if I haven't shown the results here.]

To demonstrate the combined effects of both F_{spec} >> F_{gen} biases (*a priori* and Favour Specificity), I simulate acquisition of the three sample languages using Learner D, defined with the settings in Table 20.

Parameter	**Setting**
All basic parameters	Default
A priori bias	d = 20
Favour Specificity bias	Active

Table 20. Parameter settings for Learner D.

With both of these biases applied, learning simulations for all three sample languages fail to acquire the target grammars, producing similar results to those from Learners B and C. Test results are summarized in Table 21.

Language	**Average frequency of correct outputs**
Finnish	0.3367
North Estonian	0.9000
North Seto	0.2931
Overall	0.6222

Table 21. Summary of results from simulations with Learner D.

Finnish: Table 22 shows the final ranking values for a selection of crucial constraints, after learning from simulated Finnish data. The results are not meaningfully different from those of Learner B; the grammar acquired by Learner D is more or less fully faithful.

Constraint	**Final ranking value**
Id(Bk)σ_1	134.006
Id(Bk)	114.006
$*B_2$	112.002
$*F_3 \ldots \underline{B}_5$	110.002
$*\underline{B}_5 \ldots F_3$	108.000
$*\underline{B}_5 F_3$	108.000
…	…
$*F_3 \underline{B}_5$	106.000
…	…

Table 22. Excerpt of final ranking values for Finnish after simulation with Learner D.

North Estonian: The final ranking values for a selection of crucial constraints, after learning from simulated North Estonian inputs, are shown in Table 23. The results are not meaningfully different from those of Learner C; the grammar is still lacking a full ban against vowels from set B_1.

Constraint	**Final ranking value**
Id(Bk)σ_1	128.000
$*B_2$	116.000
$*F_5 \ldots \underline{B}_2$	116.000
$*F_3$	112.000
…	…
$*F_5 \underline{B}_2$	110.000
$*\underline{B}_5 \ldots F_3$	108.000
…	…
$*B_1$	106.000
$*\underline{B}_5 F_3$	104.000
…	…
Id(Bk)	80.000
…	…

Table 23. Excerpt of final ranking values for North Estonian after simulation with Learner D. A selection of no-disagreement constraints are included here for the purpose of comparison with results of simulations discussed in subsequent sections.

North Seto: Table 24 shows the final ranking values for a selection of crucial constraints, after learning from simulated North Seto data. The results are not meaningfully different from those of Learner B; the grammar acquired by Learner D is more or less fully faithful.

Constraint	Final ranking value
Id(Bk)σ_1	136.060
Id(Bk)	116.060
*F_4…$\underline{B}_5$	116.000
*$\underline{B}_5$…F_4	116.000
*$\underline{B}_5F_4$	114.000
*$F_4\underline{B}_5$	112.000
…	…
*B_1	102.000
…	…

Table 24. Excerpt of final ranking values for North Seto after simulation with Learner D.

4.2.4. Discussion

The application of the *a priori* bias enables the learner to produce grammars in which specific faithfulness constraints are ranked higher than their general counterparts. In the context of Balto-Finnic languages, such a bias facilitates the kind of first-syllable privilege that languages in this typology require– whether to ensure that neutralization only occurs in non-initial syllables (as for North Estonian) or to allow for harmony to be driven by the first syllable (as for Finnish and North Seto).

The Favour Specificity bias has a similar effect, but is more flexible, allowing specific faithfulness constraints to rise independently of their general counterparts. However, without a minimum required distance between the specific and general faithfulness constraints, there are situations in which it results in a more crowded sequence of final ranking values (for example, North Seto Learner C). This means a more variable final grammar, in a learning context where variability is not a desired characteristic of the target grammar.

Even with the degree of success shown by the North Estonian Learners B and C, it is clear that neither of these biases is enough for successful learning of the sample languages. A particular obstacle that recurs consistently throughout the simulations discussed throughout Section 4.2 is that most of the markedness constraints do not shift away from their initial values to any great degree (Figure 1). There are a number of reasons for this, which are discussed below, but the overarching consequence is that most of the relative rankings between the various markedness constraints do not have the opportunity to become anywhere near categorical. With constantly shifting markedness pressures and steadily rising faithfulness constraints, the learner cannot determine which markedness constraints to credit with any successful outputs and is only able to start selecting the intended winners as optimal once the faithfulness constraints have surpassed the chaos of the markedness constraints. After that point, since the learners receive only positive evidence, the faithfulness constraints continue to get credit for any winners, rendering the markedness constraints effectively useless. [TODO update this once I discuss the issue from Section 4.2.1.3 with AMT.]

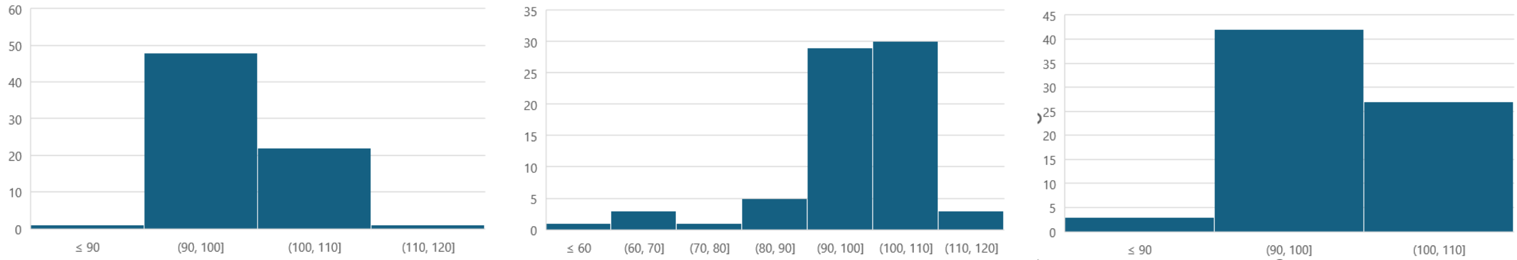

Figure 1. Distribution of final markedness constraint ranking values for Finnish, North Estonian, and North Seto Learners B. [These sad, sad plots need improving.]

Due to the nearsightedness of the GLA and the use of positive evidence only, markedness constraints that are never violated by the learning data (e.g., *$\underline{B}_5 \ldots F_3$ in Finnish) are highly unlikely to ever be violated by a generated output; the only way this would happen is due to evaluation noise. Thus they have negligible opportunity to be promoted as a result of such an error, and ideally the other markedness constraints *fall* from their initial values. However, that turns out to be unlikely as well. For example, the symmetrical properties of *B_5 and *F_5 result in these two constraints staying relatively stable relative to each other, and also fairly close to their initial value, as errors that promote one demote the other and vice versa; see (2). The rest of the context-free scale referring constraints *B_m and *F_n , while not perfectly symmetric, are antagonistic enough to result in approximately similar behaviour.

(2) Violation profile for a sample learning error with North Seto input.

/u..o/	*B_5	*F_5	…	Id(Bk)σ_1	Id(Bk)
✔ u..o	L** →		…		
☞ u..ø	*	W* ←	…		W* ←

As for the no-disagreement constraints, there is a slightly different issue at play. The vowel harmony constraints that are often violated and should be inactive in the target grammar do get demoted as errors are made in which they prefer the intended losers. But, given the complexity of this constraint set, there are other forces re-promoting these types of constraints. For example, consider *$F_5\underline{B}_5$ in Finnish. Since /i/ and /e/, both in set F_5, are transparent in Finnish, such a constraint is violated by many of the learning inputs and therefore is demoted when it contributes to an error. On the other hand, it is also promoted often enough as a side effect of updates related to other errors, that much of the downward movement is cancelled out. (3a) and (3b) show examples of such errors, assuming faithfulness to the first syllable.

(3) ERC excerpts for errors “accidentally” promoting *$F_5\underline{B}_5$ in Finnish.

	input	**candidates**	***F_5**	***B_5**	***$F_3\underline{B}_5$**	***$F_5\underline{B}_5$**

a.	/i..ø/	i..ø ~ i..o	L	W		W
b.	/æ..ø/	æ..ø ~ æ..o	L	W	W	W

In ERC (3a), both candidates are grammatical but only the intended winner is faithful to the input. Therefore when the learner selects the loser as optimal in order to avoid violating the currently-highly-ranked *F_5, the update promotes *$F_5\underline{B}_5$ even though it had nothing to do with the selection of the winner and is in fact very reasonable to violate in Finnish.

In ERC (3b), the learner again selects the loser in order to avoid violating *F_5. In this case, the resulting promotion of *$F_5\underline{B}_3$ is desired. But due to the no-disagreement constraints being built up from the nested stringency sets, the loser's violation (and resulting promotion) of *$F_5\underline{B}_3$ also necessarily means promotion of superset-referring *$F_5\underline{B}_5$.

Whether considering the context-free markedness constraints or the no-disagreement constraints, either way we run the risk of producing a strictly faithful grammar (which accounts for all of the learning data but no potential unfaithful test data) if the general faithfulness constraint is permitted to rise above the markedness constraints as their values oscillate.

The need for more space between the (ideally) higher-ranked markedness constraints and the lower ones is hindered by their oscillation, but can be facilitated by defining a learner with asymmetry between its promotion vs demotion amounts. This adaptation - a modified update rule - is presented in Section 4.3.

4.3. Promotion rate

GLA-type learners make adjustments to the ranking values after each error made by the current hypothesized grammar. While all variations on this theme agree that constraint *demotion* is necessary to the learning process, arguments have been made both for (e.g., Boersma 1997, 1998; Magri 2012) and against (e.g., Tesar & Smolensky 1998) the idea of permitting constraint *promotion* as well. I subscribe to Magri's (2012) claim that some promotion must be required in order to allow for re-ranking of faithfulness constraints which, in a learning environment that assumes faithful underlying forms for licit inputs, never prefer losers.

The amount that winner-preferring constraints get promoted might be calculated as a fraction of the current plasticity, with the fraction determined as a function of the *number* of winner-preferring and/or loser-preferring constraints at that update. E.g.,

promotion amount = (promotion rate) x plasticity

4.3.1. Rationale

At the low end, a promotion rate of 0 means that initially-low faithfulness constraints would remain stuck at their starting values; they need some way to rise to allow for adjustments to rankings as new learning inputs are encountered. At the high end, a promotion rate of 1 (or more) means that every winner-preferring constraint is given full credit for preference of the winner. However, we should consider avoiding full-fledged promotion of constraints in the case of an ERC that contains two or more constraints that prefer the intended winner, in order to avoid overpromoting when it is not clear which of those constraints should be credited with preference of the winner (Credit Problem; Dresher, 1999). Between these two extremes, different ranges of promotion rates have been shown to result in efficient convergence, inefficient convergence, or non-convergence of a GLA-type learner (Magri 2012: 265); see Figure 2.

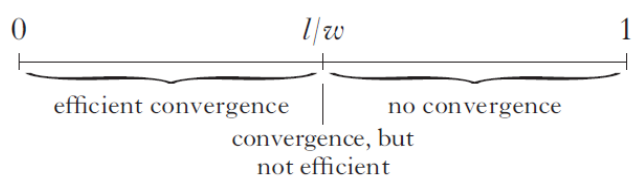

Figure 2: Magri (2012 (66)) shows how convergence and efficiency depend on promotion rate.

The goal, then, is to select a function for promotion rate ∈ (0,1) and ideally ∈ (0,L/W).[3] Various options have been tested and are detailed in 4.3.2.

[3] In this and the other references to promotion rate, W = number of winner-preferring constraints and L = number of loser-preferring constraints.

4.3.2. Implementation

At each learning update, the number of winner-preferring and loser-preferring constraints are determined from the relevant ERC. The ranking values of the loser-preferrers are decreased by the plasticity amount, and those of the winner-preferrers are increased by the promotion rate as a fraction of the plasticity. The default value for promotion rate is 1 (that is, promotion amount is equal to plasticity). I also consider four different promotion rates as functions of the number of loser and/or winner-preferring constraints; they are described below and labeled Types 1 through 4. All four of these promotion rates are $\in (0, L/W)$[4] and therefore satisfy Magri's (2012) requirements for efficient convergence.

Magri (2012) proposes a *calibrated* promotion rate:

(4) Type 1: promotion rate = $L / (1 + W)$

However, this calibrated rate has the potential to produce fractions greater than one, which flies even harder in the face of the Credit Problem as discussed in 4.3.1.

I test three additional options for a tempered promotion rate, each guaranteed to produce a fraction no greater than 1:

(5) Type 2: promotion rate = $1 / W$
(6) Type 3: promotion rate = $L / (L + W)$
(7) Type 4: promotion rate = $1 / (1 + W)$ (Magri & Kager, 2015)

4.3.3. Simulation results - promotion rate

To demonstrate the effect of a tempered promotion rate, I simulate acquisition of the three sample languages using Learner E, defined with the settings in Table 25. The choice of Type 2 for the promotion rate is for illustrative purposes and is strictly based on simplicity of the expression rather than any other factor. Results using learners with Types 1, 3, and 4 are summarized in Section 4.5.X.

Parameter	Setting
All basic parameters	Default
Promotion rate	$1 / W$ (Type 2)

Table 25. Parameter settings for Learner E.

With the modified promotion rate applied - even without the initial help of the specific-over-general faithfulness bias - the overall results are an improvement on those from earlier learners. Results are summarized in Table 26.

[4] True as long as $L \geq 1$, which is guaranteed because a candidate with no loser-preferring constraints would never have been selected as optimal.

Language	Average frequency of correct outputs
Finnish	0.9109
North Estonian	0.6264
North Seto	0.7327
Overall	0.7567

Table 26. Summary of results from simulations with Learner E.

However, inspection of the final ranking values reveal some interesting details not conveyed in these summary results.

Finnish: Table 27 shows the final ranking values for a selection of crucial constraints, after learning from simulated Finnish data. As for several of the previous learners, the key markedness constraints are at the top of the rankings, but in this case both faithfulness constraints are significantly lower than the full ban $*B_2$ and the F_3/B_5 vowel harmony constraints. This grammar does almost exactly what is needed for Finnish: it completely bans vowels in set B_2 and ensures the required harmony and transparency patterns among the remaining vowels. However, it does so without any faithfulness to the first syllable. Therefore winners are always harmonic, but whether they are back or front has is determined by either general faithfulness (the harmonic candidate with fewer faithfulness violations is preferred over a different harmonic candidate with more faithfulness violations) or, if faithfulness violations are equal, then by other lower-ranked markedness constraints (whether context-free or no-disagreement) that should not ideally be active. Tableaux (8) and (9) show examples of each of these cases.

Constraint	Final ranking value
$*B_2$	101.300
$*F_3\underline{B}_5$	100.000
$*F_3\ldots\underline{B}_5$	100.000
$*\underline{B}_5F_3$	100.000
$*\underline{B}_5\ldots F_3$	100.000
… (several other unviolated VH constraints)	… 100.000
Id(Bk)	64.897
…	…
Id(Bk)σ_1	42.658
…	…

Table 27. Excerpt of final ranking values for Finnish after simulation with Learner E.

(8) Sample evaluation of input /ø..o/ in the Finnish grammar acquired by Learner E. The grammar selects the candidate with fewer violations of $*F_1$ even though it is not faithful to the first syllable, nor should such a constraint be active in this language.

	/ø..o/	$*F_3\underline{B}_5$	$*F_3…\underline{B}_5$	$*\underline{B}_5F_3$	$*\underline{B}_5…F_3$	Id(Bk)	$*F_1$
	ø..o	*!	*				*
	ø..ø					*	*!*
	o..ø			*!	*	**	*
☞	o..o					*	

(9) Sample evaluation of input /ø..ɑ..u/ in the Finnish grammar acquired by Learner E. The grammar selects the candidate with fewer overall violations of Id(Bk) even though it is not faithful to the first syllable. Disharmonic candidates are omitted from this tableau for the sake of simplicity, since they are ruled out immediately by the highest-ranked no-disagreement constraints.

	/ø..ɑ..u/	$*F_3\underline{B}_5$	$*F_3…\underline{B}_5$	$*\underline{B}_5F_3$	$*\underline{B}_5…F_3$	Id(Bk)	$*F_1$
	ø..æ..y					*!*	*
☞	o..ɑ..u					*	

AMT HAS READ UP TO HERE

North Estonian: The final ranking values for a selection of crucial constraints, after learning from simulated North Estonian inputs, are shown in Table 28. The final grammar shows that this learner, due in part to only having access to positive evidence, has "misunderstood" North Estonian as being a vowel harmony language. With the modified promotion rate ensuring that demotions are larger than promotions, the restricted vowels in non-initial syllables are accounted for by no-disagreement constraints ($*\underline{B}_5F_3$, $*\underline{B}_5…F_3$, $*F_5\underline{B}_2$, $*F_5…\underline{B}_2$) rather than context-free constraints ($*B_2$, $*F_3$), since those no-disagreement constraints are never violated while the relevant context-free constraints are violated often by vowels in the first syllable.
TODO some ERCs maybe??
$*B_1$ does successfully implement a full ban as required, but similar to the Finnish result, the loose vowel harmony attested by this grammar is not driven by faithfulness to the first syllable; rather, it is driven by general faithfulness over all segments in the word. Even if the first-syllable faithfulness was ranked higher, interpreting North Estonian as a vowel harmony language rather than one with broader positional restrictions means that vowels in sets B_2, and F_3 are banned only when following a vowel of the opposite backness. Hence patterns such as [y..æ] are

deemed acceptable even though such sequences are not attested in North Estonian; see Tableau (10).

Constraint	**Final ranking value**
$*\underline{B}_5F_3$	100.154
$*\underline{B}_5\ldots F_3$	100.154
…	…
$*B_1$	100.000
$*F_5\underline{B}_2$	100.000
$*F_5\ldots\underline{B}_2$	100.000
…	…
Id(Bk)	75.549
…	…
Id(Bk)σ_1	53.295
…	…
$*B_2$	48.880
…	…
$*F_3$	-17.027
…	…

Table 28. Excerpt of final ranking values for North Estonian after simulation with Learner E. A selection of no-disagreement constraints are included here for the purpose of comparison with results of simulations discussed in subsequent sections.

(10) Sample evaluation of test input /y..æ/ in the North Estonian grammar acquired by Learner E. The grammar selects the candidate in which both vowels are front, even though it has a vowel in set F_3 in a non-initial syllable.

	/y..æ/	$*\underline{B}_5F_3$	$*\underline{B}_5\ldots F_3$	Id(Bk)	$*F_3$
☞	y..æ				**
	y..ɑ			*!	*
	u..æ	*!	*	*	*
	u..ɑ			*!*	

North Seto: Table 29 shows the final ranking values for a selection of crucial constraints, after learning from simulated North Seto data. The key vowel harmony constraints are successfully at the top of the rankings, ensuring the required harmony and transparency patterns. The positional restriction $*B_1$ is also below Id(Bk)σ_1 as it should be; however, because the general faithfulness constraint is above the specific one, it is true both that (a) winners are always harmonic but their backness is determined by general faithfulness (the harmonic candidate with fewer faithfulness violations is preferred over a different harmonic candidate with more faithfulness violations) before first-syllable faithfulness and (b) $*B_1$ is too far down to have the opportunity to be active, so there is in effect no restriction on non-initial syllables. Tableaux (11) and (12) show examples of each of these cases.

Constraint	**Final ranking value**
$*\underline{B}_5F_4$	100.286
$*\underline{B}_5 \ldots F_4$	100.286
$*F_4\underline{B}_5$	100
$*F_4 \ldots \underline{B}_5$	100
…	…
Id(Bk)	70.300
…	…
Id(Bk)σ_1	50.988
…	…
$*B_1$	43.978

Table 29. Excerpt of final ranking values for North Seto after simulation with Learner E.

(11) Sample evaluation of input /ɯ..ɯ/ in the North Seto grammar acquired by Learner E. Since the input is already harmonic, the grammar selects the candidate with the fewest faithfulness violations, even though [ɯ] is restricted and should not appear in the second syllable.

	/ɯ..ɯ/	$*\underline{B}_5F_4$	$*\underline{B}_5 \ldots F_4$	$*F_4\underline{B}_5$	$*F_4 \ldots \underline{B}_5$	Id(Bk)	Id(Bk)σ_1	$*B_1$
☞	ɯ..ɯ							**
☹	ɯ..i					*!		*
	i..ɯ					*!	*	*
	i..i					*!*	*	

(12) Sample evaluation of input /æ..ɯ..ɤ/ in the North Seto grammar acquired by Learner E. The grammar selects the candidate with fewest overall violations of Id(Bk) even though it both contains a non-initial [ɯ] and is not faithful to the first syllable. Disharmonic candidates are omitted from this tableau for the sake of simplicity, since they are ruled out immediately by the highest-ranked no-disagreement constraints.

	/æ..ɯ..ɤ/	$*\underline{B}_5F_4$	$*\underline{B}_5 \ldots F_4$	$*F_4\underline{B}_5$	$*F_4 \ldots \underline{B}_5$	Id(Bk)	Id(Bk)σ_1	$*B_1$
☹	æ..i..e					**!		
☞	ɑ..ɯ..ɤ					*	*	*
	ɑ..i..ɤ					**!	*	

4.3.4. Simulation results - F_{spec} >> F_{gen}, promotion rate

To demonstrate the combined effects of the *a priori* bias, Favour Specificity bias, and tempered promotion rate, I simulate acquisition of the three sample languages using Learner F, defined with the settings in Table 30.

Parameter	Setting
All basic parameters	Default
A priori bias	*d* = 20
Favour Specificity bias	Active
Promotion rate	1 / *W* (Type 2)

Table 30. Parameter settings for Learner F.

With all three of these modifications implemented, the results show improvement yet again compared to those from prior learners. Results are summarized in Table 31.

Language	Average frequency of correct outputs
Finnish	1.0000
North Estonian	0.8853
North Seto	0.9997
Overall	0.9617

Table 31. Summary of results from simulations with Learner F.

At this point, the grammar acquired by the Finnish learner is achieving 100% success on tests, the North Seto grammar nearly so, and the North Estonian grammar has improved significantly as compared to the one acquired by Learner E (but not D???). Final rankings are presented and analyzed below.

Finnish: Table 32 shows the final ranking values for a selection of crucial constraints, after learning from simulated Finnish data. This grammar meets all the requirements for a target Finnish grammar (TODO reminder of Hasse diagram), and the ranking values are far enough apart in value to behave effectively categorically, as evidenced by the 1.0 frequency of correct outputs during testing.

Constraint	Final ranking value
$*B_2$ …	100.800 …

$*F_3\underline{B}_5$ $*F_3…\underline{B}_5$ $*\underline{B}_5F_3$ $*\underline{B}_5…F_3$ … Id(Bk)σ_1 Id(Bk)	100.000 100.000 100.000 100.000 … 68.961 43.483

Table 32. Excerpt of final ranking values for Finnish after simulation with Learner F.

North Estonian: The final ranking values for a selection of crucial constraints, after learning from simulated North Estonian inputs, are shown in Table 33. These rankings improve those of Learner D in that $*B_1$ is at the top, implementing a full ban. They also improve those of Learner E in that Id(Bk)σ_1 outranks Id(Bk), ensuring faithfulness to the vowel in the first syllable. However, although $*B_2$ is sandwiched between the two faithfulness constraints as required, the “North Estonian as a vowel harmony language” problem still persists with highly ranked $*\underline{B}_5F_3$, $*\underline{B}_5…F_3$, $*F_5\underline{B}_2$, and $*F_5…\underline{B}_2$, but $*F_3$ below general faithfulness and therefore inactive (though Id(Bk) and $*F_3$ are close enough in ranking value to swap places some of the time due to evaluation noise). Tableaux (13) and (14) show how vowels in set B_2 are appropriately restricted in non-initial syllables but those in set $*F_3$ may not be.

Constraint	**Final ranking value**
$*B_1$ $*F_5\underline{B}_2$ $*F_5…\underline{B}_2$ $*\underline{B}_5F_3$ $*\underline{B}_5…F_3$ … Id(Bk)σ_1 … $*B_2$ … Id(Bk) $*F_3$ …	100.000 100.000 100.000 100.000 100.000 … 70.975 … 60.8 … 39.148 37.626 …

Table 33. Excerpt of final ranking values for North Estonian after simulation with Learner F. A selection of no-disagreement constraints are included here for the purpose of comparison with results of simulations discussed in subsequent sections.

(13) Sample evaluation of test input /ɑ..ɤ/ in the North Estonian grammar acquired by Learner F. The grammar successfully selects the candidate that avoids [ɤ] in the second syllable.

	/ɑ..ɤ/	$*F_5\underline{B}_2$	$*F_5…\underline{B}_2$	$*\underline{B}_5F_3$	$*\underline{B}_5…F_3$	Id(Bk)σ_1	$*B_2$	Id(Bk)	$*F_3$

	ɑ..ɤ						*!		
☞	ɑ..e							*	
	æ..ɤ	*!	*			*	*	*	
	æ..e					*!		**	

(14) Sample evaluation of test input /y..ø/ in the North Estonian grammar acquired by Learner E. The grammar selects the candidate in which both vowels are front, even though it has a vowel in set F_3 in a non-initial syllable.

	/y..ø/	*$F_5\underline{B}_2$	*$F_5\ldots\underline{B}_2$	*$\underline{B}_5F_3$	*$\underline{B}_5\ldots F_3$	Id(Bk)σ_1	*B_2	Id(Bk)	*F_3
☞	y..ø								**
☹	y..o							*!	*
	u..ø			*!	*	*		*	*
	u..o					*!		**	

North Seto: Table 34 shows the final ranking values for a selection of crucial constraints, after learning from simulated North Seto data. This grammar meets nearly all the requirements for a target North Seto grammar (TODO reminder of Hasse diagram), though some pairs of ranking values (TODO IdBkSyl1 vs *B1; *B1 vs *B1F5… which honestly shouldn't be in there anyway) are just close enough together that evaluation noise results in the odd swapped ranking; hence the 0.9997 rate of correct outputs during testing.

Constraint	**Final ranking value**
*$F_4\underline{B}_5$	100.000
*$F_4\ldots\underline{B}_5$	100.000
*$\underline{B}_5F_4$	100.000
*$\underline{B}_5\ldots F_4$	100.000
Id(Bk)σ_1	70.289
*B_1	62.380
*$\underline{B}_1F_5$	54.000
Id(Bk)	44.219
…	…

Table 34. Excerpt of final ranking values for North Seto after simulation with Learner F.

4.3.5. Discussion

Modifying the update rule by adjusting the learning algorithm's promotion rate serves to create space between markedness constraints by ensuring that any oscillating constraint values oscillate on an overall downward (at least, in this case) trajectory. For a constraint set such as this one, where the use of stringency sets produces antagonistic - or near-antagonistic - constraints, this facilitates the differentiation of constraints whose values stay relatively stable due to never being violated by the positive-evidence learning inputs (such as TODO) versus constraints whose values stay relatively stable as a result of oscillation due to repeated violation (such as TODO).

The effects of the tempered promotion rate can be understood via comparison of either Learner A with Learner E, or Learner C with Learner F. By inspecting the final rankings of those learners (for any language; Table 35 summarizes results table references), it is clear that the effect on the final rankings of modifying the promotion rate in this way, is for markedness constraints to be high enough that they can be active in accounting for the patterns attested in the input data, rather than leaving that work entirely to the faithfulness constraints. This is facilitated by virtue of the markedness constraints' relative rankings becoming clearly articulated enough to reduce the error rate, before faithfulness constraints rise so far as to overcome the markedness constraints.

Learner	**Settings**	**Finnish results table**	**North Estonian results table**	**North Seto results table**
A	default	4	5	6
E	default + promotion rate	27	28	29
C	default + *a priori* + Favour Specificity	16	17	18
F	default + *a priori* + Favour Specificity + promotion rate	32	33	34

Table 35. TODO.

For the vowel harmony languages, Finnish and North Seto, Learner F achieves near-perfect results. However, for North Estonian with its positional restrictions, Learner F acquires a grammar that accounts for the patterns attested in the positive-evidence learning inputs via no-disagreement constraints rather than context-free markedness constraints, since these particular no-disagreement constraints are never violated while the context-free constraints are part of the antagonistic collection of constraints that oscillate downward as they are repeatedly violated by the learning data. The problem with such a result is that the no-disagreement constraints are too specific– they account perfectly well for the input data but fail to generate correct outputs when given ungrammatical test data. The vowels in non-initial syllables need to be restricted not only when they are disharmonic with the vowels in initial syllables, but also even when they harmonize. This is in essence a restrictiveness problem: in the same way that prioritizing specific faithfulness constraints over general ones allows for a more restrictive grammar, it is clear here that general markedness constraints need to be prioritized over specific ones in order to ensure better restrictiveness in terms of markedness as well.

There is a wide range of potential strategies for prioritizing generality in markedness constraints; Section 4.4 proposes several options and presents results of one specific implementation.

4.4. M_{gen} >> M_{spec}

My constraint set does not have any explicitly defined pairs of specific vs general markedness constraints like it does faithfulness constraints. However, it is nevertheless possible to determine the relative generality of various pairs of markedness constraints, a task that is made easier due to the fact that all of the markedness constraints refer to the same stringency scales. Using a general-over-specific markedness bias works toward the same goal as the specific-over-general faithfulness bias: learning a grammar that is as restrictive as possible.

4.4.1. Rationale

The learner has difficulty learning a correct ranking for Northern Estonian because the vowel harmony (no-disagreement) constraints do a better job of explaining the limited vowels in non-initial syllables in the learning data than the context-free segmental constraints do. Although the context-free constraints are sometimes violated (specifically, by vowels in the first syllable), in fact they are much more general and can better deal with ungrammatical inputs than the no-disagreement constraints can. The more general constraints make the grammar more restrictive, which is useful for avoiding overgeneration when encountering ungrammatical inputs.

The rationale for a general-over-specific markedness bias is to give the most general markedness constraints an opportunity to get credit for the phonotactics of the target grammar, in order to ensure maximal restrictiveness.

The preference for more general markedness constraints is not persistent; rather, it is implemented as an initial articulated hierarchy of markedness constraint values, calculated as a function of each constraint's rate of application in a sample set of inputs (cf. Albright & Hayes, 2006, p. 11 TODO - talk about junk constraints) and can be freely reversed by learning data.

4.4.2. Implementation

Within the scale-referring markedness constraints central to this project, there are several dimensions on which generality can be measured, informed by set theory. Using these dimensions, one can determine the relative place of various markedness constraints in the initial ranking values.

- Dimension 1: size of stringency set. E.g., $*B_5$ is more general than $*B_2$, because it bans the whole class of back vowels rather than just two of them.
- Dimension 2: context-sensitivity. E.g., $*B_5$ is more general than $*\underline{B}_5F_2$ and $*\underline{B}_5\ldots F_2$, because it bans all instances of back vowels, not just when they precede a front vowel.
- Dimension 3: size of stringency set in context. E.g., $*\underline{B}_5F_5$ is more general than $*\underline{B}_5F_2$, because it bans back vowels followed by any of the five front vowels rather than by either of just two of them.
- Dimension 4: scope of application: E.g., $*\underline{B}_5\ldots F_2$ is more general than $*\underline{B}_5F_2$, because it bans all sequences including an earlier back vowel from set B_5 and a later front vowel from set F_2, not just when they are adjacent.

Using these dimensions, it is straightforward to establish dozens of generality-based hierarchies of markedness constraints. However, it is very difficult to determine how the constraints in those separate hierarchies should interleave in order to create an overall initial distribution of markedness constraints based on generality. For example, $*B_5$ is more general than $*B_2$, and $*B_5$ is more general than $*\underline{B}_5F_2$, but the relationship between $*B_2$ and $*\underline{B}_5F_2$ is not clear.

Given the difficulty of calculating the overall distribution of markedness based on their set-theoretic relationships, I propose instead four different methods for the learner to determine the initial relative rankings, detailed in Sections 4.4.2.1 through 4.4.2.4 below.

4.4.2.1. Constant distribution function

Distribution Function 1 ($\mathcal{F}_1$): The first approach is to assume that all markedness constraints have the same starting value. This is a baseline that is not informed by relative generality at all. Within $\mathcal{F}_1$, I test starting values of both 100 and 300.

4.4.2.2. Stratified distribution function - by constraint type

Distribution Function 2 ($\mathcal{F}_2$): The second approach is to construct discrete strata of markedness constraints, based on their level of generality from the perspective of constraint type: context-free, long-distance no-disagreement constraints, and local no-disagreement constraints.

The strata are assigned by $\mathcal{F}_2$ as follows:

- Stratum 1 contains all context-free markedness constraints. For example, $*F_1$ and $*B_3$.
- Stratum 2 contains all long-distance no-disagreement constraints. For example, $*\underline{B}_2\ldots F_5$ and $*F_1\ldots\underline{B}_2$.
- Stratum 3 contains all local no-disagreement constraints. For example, $*F_4\underline{B}_2$ and $*\underline{B}_5F_5$.

The initial ranking values for the strata must be specified as well. I use 140, 120, and 100 for the first, second, and third strata, respectively.

4.4.2.3. Stratified distribution function - by stringency set

Distribution Function 3 ($\mathcal{F}_3$): The third approach is to construct discrete strata of markedness constraints, based on their level of generality from the perspective of the cardinality of the sets referred to by each constraint. Because all markedness constraints in this project are scale-referring, it is straightforward to use the front and back vowel sets to assign strata.

In the top-down version of $\mathcal{F}_3$ (called $\mathcal{F}_{3t}$), I assign strata greedily, as follows:

- Stratum 1 (the highest) contains all markedness constraints whose largest component refers to set B_5 or F_5. For example, $*B_5$, $*\underline{B}_3F_5$, and $*F_1\ldots\underline{B}_5$.
- Stratum 2 contains all markedness constraints whose largest component refers to set F_4 (recall that B_4 is undefined). For example, $*F_4$ and $*\underline{B}_1\ldots F_4$.
- Stratum 3 contains all markedness constraints whose largest component refers to set B_3 or F_3.
- Stratum 4 contains all markedness constraints whose largest component refers to set B_2 (recall that F_2 is undefined).
- Stratum 5 (the lowest) contains all of the remaining markedness constraints, which are the ones that don't refer to any sets larger than B_1 or F_1.

In the bottom-up version of $\mathcal{F}_3$ (called $\mathcal{F}_{3b}$), I assign strata greedily, as follows:

- Stratum 5 (the lowest) contains all markedness constraints whose smallest component refers to set B_1 or F_1. For example, $*F_1$, $*\underline{B}_1F_4$, and $*\underline{B}_3\ldots F_1$.
- Stratum 4 contains all markedness constraints whose smallest component refers to set B_2 (recall that F_2 is undefined). For example, $*B_2$ and $*\underline{B}_2F_5$.
- Stratum 3 contains all markedness constraints whose smallest component refers to set B_3 or F_3.
- Stratum 2 contains all markedness constraints whose smallest component refers to set F_4 (recall that B_4 is undefined).
- Stratum 1 (the highest) contains all of the remaining markedness constraints, which are the ones that don't refer to any sets smaller than B5 or F5.

The initial ranking values of the strata are specified as 180, 160, 140, 120, and 100 for the first through fifth strata, respectively.

4.4.2.4. Continuous distribution function

Distribution Function 4 ($\mathcal{F}_4$): The fourth approach is to calculate the initial ranking values of markedness constraints via a function with a more finely distributed range. In this case, as mentioned at the beginning of section 4.4.2, it is very difficult to use the theoretical relationships between different types of constraints in order to determine the relative generality of individual pairs of constraints. Instead, I use a numerical method based on the application rate of each constraint within the inputs seen by the learner.

Recall that in Section 4.1.1 I present the idea of the learning process taking place over four stages, each with a declining plasticity and consisting of 5000 learning trials. For the implementation of this particular bias, I prepend a pre-learning "observation" stage, with plasticity = 0.

During the observation stage, the learner is fed randomly-sampled inputs just as it is during the learning stages. However, rather than using the current hypothesized grammar to compare the optimal output to the intended winner, the learner simply observes how many times each markedness constraint is violated by the candidate corresponding to the heard input, and adds it to the tally for that constraint. Once the learner has heard all of the inputs in this stage, the violations tally for each constraint is divided by the number inputs heard, producing an average generality measure (application rate) for each markedness constraint.

Before the first learning stage, the application rate is used to calculate the initial ranking value for each markedness constraint, modifying it from the default value of 100. The calculation of this distribution requires two additional parameters; these determine:

a) The initial ranking value corresponding to a constraint with a 0.0 application rate. This parameter is referred to as the *y-intercept coefficient*. The tested values for the y-intercept coefficient include 0.5, 1.0, and 1.5.
b) The initial ranking value corresponding to a constraint with a 1.0 application rate. This parameter is referred to as the *slope coefficient*. The tested values for the slope coefficient include 0.5, 1.0, and 1.5.

Then the initial ranking value for each markedness constraint M is calculated using the following equation:

$\theta_{\text{Minit}} = 100(b + mg_M)$,

where b = y-intercept coefficient

m = slope coefficient

g_M = application rate for constraint M

4.4.2.5. Random distribution function

Distribution Function 5 ($\mathscr{F}_5$): In addition to having a constant distribution function $\mathscr{F}_1$ as a reference point against which to consider the success of the other, generality-based markedness distribution functions, I also ran simulations with markedness constraints randomly distributed over similar intervals as for $\mathscr{F}_4$.

The initial ranking value for each markedness constraint M is calculated using the following equation:

$\theta_{\text{Minit}} = 100(b + mr_M)$,

where b = y-intercept coefficient (0.5, 1.0, or 1.5)

m = slope coefficient (0.5, 1.0, or 1.5)

r_M = simulated application rate for constraint M, randomly sampled from [0,1]

4.4.3. Simulation Results - M_{gen} >> M_{spec}

To demonstrate the effect of a general-over-specific markedness bias, I simulate acquisition of the three sample languages using Learner G, defined with the settings in Table 35. The choice of $\mathscr{F}_4$ for the distribution function is for illustrative purposes and was selected for its minimal requirement of *a priori* knowledge or calculation on the part of the learner, given that the initial distribution of markedness constraints can be determined based strictly on observation rather than prior analysis of set-theoretic relationships between classes of markedness constraints. Results using learners with $\mathscr{F}_1$, $\mathscr{F}_2$, $\mathscr{F}_3$, and $\mathscr{F}_5$ (along with other values of *b* and *m*) are summarized in Section 4.5.X.

Parameter	Setting
All basic parameters	Default
Initial markedness values	$\mathscr{F}_4$ (continuous distribution), with $b = 1.0$, $m = 1.0$

Table 35. Parameter settings for Learner G.

With only the general-over-specific markedness bias applied, the results do not initially appear to be any better than those of Learner A (default settings). Results are summarized in Table 36.

Language	Average frequency of correct outputs
Finnish	0.2618
North Estonian	0.2455
North Seto	0.2982
Overall	0.2685

Table 36. Summary of results from simulations with Learner G.

However, a closer inspection of the ranking values for each language reveals that some significant changes are in fact present, even though their effects are masked in these summary results.

Finnish: Table 37 shows the final ranking values for a selection of crucial constraints, after learning from simulated Finnish data. Due to Learner G's lack of any of the other biases discussed thus far, the grammar acquired by the Finnish learner is simply ruled by a high-ranking general faithfulness constraint. Subsidiary to Id(Bk), however, note that both the markedness constraints *and* Id(Bk)σ_1 are ranked correctly relative to each other. This is contrary to Finnish Learner A, in whose final grammar Id(Bk)σ_1 was ranked lower than both the full ban *B_2 and the vowel harmony constraints *$F_3\underline{B}_5$, *$F_3\ldots\underline{B}_5$, *$\underline{B}_5F_3$, and *$\underline{B}_5\ldots F_3$. Id(Bk)$\sigma_1$'s relative height in the rankings shown in Table 37 is due to the learner's opportunity to make plenty of errors in the first syllable, facilitated by the high initial value of more general markedness constraints. TODO an ERC?

Constraint	Final ranking value
Id(Bk)	254.000

…	…
$*B_2$	228.000
…	…
$Id(Bk)\sigma_1$	178.000
…	…
$*F_3\underline{B}_5$	100.000
$*F_3\ldots\underline{B}_5$	100.000
$*\underline{B}_5F_3$	100.000
$*\underline{B}_5\ldots F_3$	100.000
…	…

Table 37. Excerpt of final ranking values for Finnish after simulation with Learner G.

North Estonian: The final ranking values for a selection of crucial constraints, after learning from simulated North Estonian inputs, are shown in Table 38. Similar to the Finnish results, Id(Bk) is still at the very top. But contrary to several of the previous learners, the final grammar acquired by Learner G has the more general context-free markedness constraints $*F_3$ and $*B_2$ ranked significantly higher than the more specific no-disagreement constraints $*F_5\underline{B}_2$, $*F_5\ldots\underline{B}_2$, $*\underline{B}_5F_3$, and $*\underline{B}_5\ldots F_3$, which shows promise in terms of the potential for positional restrictions being enforced regardless of harmony.

Constraint	**Final ranking value**
Id(Bk)	254.000
…	…
$*F_3$	216.840
$*B_2$	204.840
…	…
$*B_1$	160.000
$Id(Bk)\sigma_1$	154.000
…	…
$*F_5\underline{B}_2$	100.000
$*F_5\ldots\underline{B}_2$	100.000
$*\underline{B}_5F_3$	100.000
$*\underline{B}_5\ldots F_3$	100.000
…	…

Table 38. Excerpt of final ranking values for North Estonian after simulation with Learner G. A selection of no-disagreement constraints are included here for the purpose of comparison with results of simulations discussed in subsequent sections.

North Seto: Table 39 shows the final ranking values for a selection of crucial constraints, after learning from simulated North Seto data. Of the three sample languages, the final grammar learned by the North Seto learner is the only one that the general-over-specific markedness bias does not appear to affect in any potentially-useful way.

Constraint	**Final ranking value**

Id(Bk)	254.000
…	…
Id(Bk)σ_1	210.000
…	…
*B_1	167.500
…	…
*$F_4\underline{B}_5$	100.000
*$F_4 \ldots \underline{B}_5$	100.000
*$\underline{B}_5 F_4$	100.000
*$\underline{B}_5 \ldots F_4$	100.000
…	…

Table 39. Excerpt of final ranking values for North Seto after simulation with Learner G.

4.4.4. Simulation Results - F_{spec} >> F_{gen}, promotion rate, M_{gen} >> M_{spec}

To demonstrate the combined effects of the *a priori* bias, Favour Specificity bias, tempered promotion rate, and general-over-specific markedness bias, I simulate acquisition of the three sample languages using Learner H, defined with the settings in Table 40.

Parameter	**Setting**
All basic parameters	Default
A priori bias	d = 20
Favour Specificity bias	Active
Promotion rate	1 / W (Type 2)
Initial markedness values	$\mathscr{F}_4$ (continuous distribution), with b = 1.0, m = 1.0

Table 40. Parameter settings for Learner H.

Implementing all four of these modifications produces exceptional results, with even the North Estonian grammar now generating correct outputs in nearly every test evaluation. Results are summarized in Table 41.

Language	**Average frequency of correct outputs**
Finnish	1.0000
North Estonian	0.9995
North Seto	1.0000
Overall	0.9998

Table 41. Summary of results from simulations with Learner H.

Finnish: Table 42 shows the final ranking values for a selection of crucial constraints, after learning from simulated Finnish data. This grammar meets all the requirements for a target Finnish grammar (TODO reminder of Hasse diagram), and the ranking values are far enough apart in value to behave effectively categorically, as evidenced by the 1.0 frequency of correct outputs during testing.

Constraint	Final ranking value
$*B_2$	133.800
…	…
$*F_3\underline{B}_5$	100.000
$*F_3\ldots\underline{B}_5$	100.000
$*\underline{B}_5F_3$	100.000
$*\underline{B}_5\ldots F_3$	100.000
$Id(Bk)\sigma_1$	91.089
Id(Bk)	57.396

Table 42. Excerpt of final ranking values for Finnish after simulation with Learner H.

North Estonian: The final ranking values for a selection of crucial constraints, after learning from simulated North Estonian inputs, are shown in Table 43.This grammar meets all the requirements for a target North Estonian grammar (TODO reminder of Hasse diagram), though some pairs of ranking values (TODO IdBkSyl1 vs *F1; *F1 vs *F1b3… which honestly shouldn't be in there anyway) are just close enough together that evaluation noise results in the odd swapped ranking; hence the 0.9995 rate of correct outputs during testing.

Constraint	Final ranking value
$*B_1$	116.400
$*F_5\underline{B}_2$	100.000
$*F_5\ldots\underline{B}_2$	100.000
$*\underline{B}_5F_3$	100.000
$*\underline{B}_5\ldots F_3$	100.000
…	…
$Id(Bk)\sigma_1$	91.871
…	…
$*F_1$	82.841
$*B_2$	82.799
$*F_1\underline{B}_3$	76.056
…	…
$*F_3$	67.941
…	…
Id(Bk)	41.541
…	…

Table 43. Excerpt of final ranking values for North Estonian after simulation with Learner H. A selection of no-disagreement constraints are included here for the purpose of comparison with results of simulations discussed in subsequent sections.

North Seto: Table 44 shows the final ranking values for a selection of crucial constraints, after learning from simulated North Seto data. This grammar meets all the requirements for a target North Seto grammar (TODO reminder of Hasse diagram), and the ranking values are far enough apart in value to behave effectively categorically, as evidenced by the 1.0 frequency of correct outputs during testing.

Constraint	Final ranking value
$*F_4\underline{B}_5$	100.000
$*F_4\ldots\underline{B}_5$	100.000
$*\underline{B}_5F_4$	100.000
$*\underline{B}_5\ldots F_4$	100.000
…	…
$Id(Bk)\sigma_1$	99.386
$*B_1$	89.3808
…	…
Id(Bk)	56.604
…	…

Table 44. Excerpt of final ranking values for North Seto after simulation with Learner H.

STOP HERE FOR NOW PLEASE :)

4.4.5. Discussion

TODO - Summarize how adjustments to the learning parameters contributed to the success or failure of the learning simulations.

4.5. A set of ideal learners

Sections 4.2, 4.3, and 4.4 explored potential settings for the additional parameters and biases involved in addressing the various challenges encountered by an algorithmic learner with the basic settings described in Section 4.1. In Section 4.5 I take a broader view and generalize the range of settings for each parameter that produce the best overall results. I do so by examining the best-performing learners (those above a particular benchmark) and determining which parameter settings are most common in those learners. I also investigate the lowest-performing learners with those parameter settings, to ensure that the parameter settings I deem crucial to success do - at least to some degree - guarantee good results. A full summary of the results referenced in Section 4.5 is available in Appendix TODO.

4.5.1. Learners with 99.98% per-language frequency of correct results

There are six learners whose final grammars had a frequency of at least 0.9999 correct results for all three sample languages. The most common parameter settings for these learners (including the ideal learner described in Section 4.2) include:

(TODO)

- Promotion rate of Type 2 (1 / *W*) or Type 3 (*L* / (*L* + *W*))
- *A priori* bias of size 30
- Favour Specificity active
- Markedness constraints with initial values determined by the continuous distribution function $\mathcal{F}_4$, with b = 150 and m = 50

The worst-performing learner with any combination of these values produced an average frequency of correct results equal to 0.9860 (with 1.0000 for Finnish, 0.9581 for North Estonian, and 0.9999 for North Seto). This suggests that choosing settings that align with those in (TODO) defines a learner that is likely to produce excellent results.

Asdf TODO

4.5.2. Learners with 99% per-language frequency of correct results

If the benchmark for success is loosened slightly, then there are forty-eight learners whose final grammars had a frequency of at least 0.99 correct results for all three sample languages. The most common parameter settings for these learners (including the ideal learner described in Section 4.2) include:

(TODO)

- Promotion rate of Type 2 (1 / *W*), Type 3 (*L* / (*L* + *W*)), or Type 4 (1 / (1 + *W*))
- *A priori* bias of size 20, 30, or 40
- Favour Specificity active
- Markedness constraints with initial values determined by the continuous distribution function $\mathcal{F}_4$, with $b \in \{100, 150\}$ and $m \in \{50, 100\}$

The worst-performing learner with any combination of these values produced an average frequency of correct results equal to 0.8697 (with 0.7091 for Finnish, 0.8999 for North Estonian, and 1.0000 for North Seto). This suggests that choosing settings that align with those in (TODO) defines a learner that is likely to produce reasonably good results.

4.7.3. Summary and discussion

The learning parameters investigated in this chapter address a range of factors contributing to the success (or lack of it) of a learning simulation. The parameters and settings that produce the overall best results for the sample languages of Finnish, North Estonian, and North Seto are restated in (TODO) and discussed by turn below.

(TODO)

- Promotion rate of Type 2 ($1 / W$), Type 3 ($L / (L + W)$), or Type 4 ($1 / (1 + W)$)
- *A priori* bias of size 20, 30, or 40
- Favour Specificity active
- Markedness constraints with initial values determined by the continuous distribution function ($\mathcal{F}_4$), with $b \in \{100, 150\}$ and $m \in \{50, 100\}$

Promotion rate: The Type 2, Type 3, and Type 4 promotion rates all fall in the interval (0, L/W) that Magri (2012) showed to result in efficient convergence of a GLA-type learner. These promotion rates also share a second important characteristic, which is that they are guaranteed to produce fractions ≤ 1. This is crucial for ensuring that promotion amounts are no greater than the base plasticity, in order to remain conservative when giving credit to winner-preferring constraints (Credit Problem; Dresher 1999). TODO

A priori bias: The *a priori* bias sizes of 20, 30, and 40 mean that each specific faithfulness constraint has a value high enough above its general counterpart such that their relationship during evaluation is effectively categorical. This ensures that any vowels in privileged positions (in this case, the initial syllable) are in fact more likely to remain faithful to their underlying values than vowels elsewhere in the word. For the Balto-Finnic languages, this is absolutely necessary as full contrast (with respect to vowels) in any given language is only guaranteed in the first syllable. On its own, the *a priori* bias inevitably results in each pair of faithfulness constraints moving in lockstep at a distance of 20 (or 30 or 40) apart, since there will always be at least as many faithfulness errors made in the word as in the first syllable.

Favour Specificity: An active Favour Specificity bias works in concert with the *a priori* bias to prioritize specific faithfulness constraints over their general counterparts. This particular setting is implemented via the learner promoting only the specific version of a faithfulness constraint when adjusting after an error made in which both the specific and general versions are eligible to be promoted. Such a bias comes with the potential for each specific faithfulness constraint to rise further above its general counterpart than the *a priori* bias, since it will not always be the case that all faithfulness errors result in promotion of the general constraint. In the Balto-Finnic languages, this kind of additional space is important for languages with positional restrictions, which require context-free markedness constraints against particular sets of vowels to be ranked between specific and general faithfulness constraints.

$M_{gen} >> M_{spec}$: Markedness constraints with initial values distributed by the continuous distribution function $\mathcal{F}_4$, learned from input forms, are important WHY. TODO

TODO

TODO

- [ ] Double-check results summaries & top-performing parameter combinations once all sims & 100-sample tests have been run.
- [ ] Consider running second rounds of best-performing sims, to confirm first round of results.

References for Chapter 4/5

- [ ] Albright & Hayes 2006
- [ ] Angluin 1980
- [ ] Baker 1979
- [ ] Boersma 1997
- [ ] Boersma 1998
- [ ] Boersma & Hayes 2001
- [ ] Dresher 1999
- [ ] Gnanadesikan 1995
- [ ] Hayes 2004
- [ ] Hayes et al 2013 (OTSoft)
- [ ] Hayes et al 2013 (OTSoft manual v2.5)
- [ ] Jesney and Tessier (2011)
- [ ] Magri 2012
- [ ] Magri & Kager 2015
- [ ] Prince and Tesar (2004)
- [ ] Smolensky 1996
- [ ] Tesar & Smolensky 1998
- [ ] Tessier 2007

Things I need to mention/introduce/define elsewhere

- [ ] Define acronym "OT"
- [ ] Discuss undefined F2 & B4 when introducing stringency scales in ch 3
- [ ] When introducing the GLA in general (Ch 3)
    - [ ] define ranking values, ERCs, loser/winner-preferring constraints, assumption of idempotence (Magri?)
    - [ ] Also define notation for ranking values. Magri uses theta…?
    - [ ] discuss final ranking values vs stochastic evaluation / variability
    - [ ] Discuss how test phase is performed (and analyzed??). Include an explanation somewhere before this chapter re method of calculating "average frequency of correct results." Initial draft below.

- [ ] Align any references to “crucial rankings” in this chapter, with the ones discussed in the "analysis" section of ch 3
- [ ] Originally I’d written some content about the “ideal” sims in 4.2.2 re in what ways the stochastic element was causing some ungrammatical variation. This has been removed and pasted below for the time being– where should it eventually go?
- [ ] Methods for running sims (see stub below), and Python implementation of GLA in an appendix
- [ ] What kinds of testing was done / how to identify data from 100-sample vs 5-sample tests
- [ ] Overall results table in an appendix (columns: M dist type, magri type, fs, a priori size, overall results, Fin results, NEst results, NSeto results)

<u>Method of calculating “average frequency of correct results”</u>
Once a simulation has been completed, the grammar it has produced must be tested for its ability to generate all and only the grammatical forms in the language of the learning data. When a grammar G is tested, each input is evaluated 100 times and the frequency of each output recorded. In a stochastic context such as this one, there is inherent potential for the outputs to vary across evaluation instances. For example, the results for input /ø..ø/ in a grammar learned from North Estonian data might appear as in Table TODO.

/ø..ø/	Output frequency
ø..ø	0.01
ø..o	0.99
o..ø	0.00
o..o	0.00

Table TODO. Sample output frequencies for input /ø..ø/ when testing a North Estonian grammar.

Each output’s potential correctness is assessed based on two factors:

(1) Whether the initial vowel is identical to that of the input (or the initial vowel in the input was not in the language’s inventory).
(2) Whether the output is phonotactically licit in the language.

In the case of the example in Table TODO, output ø..o is correct for input /ø..ø/ but output ø..ø is not. Therefore the frequency of correct outputs for input /ø..ø/ is 0.99.

The mean of the frequency of correct outputs across all inputs in the language is taken as a measure of the overall appropriateness of the learned grammar, and reported below as “Average frequency of correct outputs.”

Stochastic element causes some ungrammatical variation

4.2.2.1 - North Estonian

These final ranking values produce a grammar that, interpreted categorically, exhibits the crucial rankings of *B1 >> Id(Bk)syl1 >> *F3, *B2 >> Id(Bk). However, given the stochastic element in the evaluation stage, there are a few pairs of constraints whose relative ranking values are close enough so as to have resulted in the occasional swapped ordering, producing a small amount of (ungrammatical) variation.

In the simplest version of a North Estonian grammar, Id(Bk)syl1 >> *F3 is the ranking that does the work of ensuring that there are no marked front vowels in non-initial syllables. However, in the grammar learned through this simulation, there are some other constraints whose final ranking values are in the interval between the values of Id(Bk)syl1 and *F3:

Id(Bk)syl1	115.938
*F1	105.560
…	…
*F1B3	97.120
…	…
*F3	72.392

Two pairs of these constraints are of interest: *F1 vs *F1B3, and Id(Bk)syl1 vs *F1.

As shown in Section 3.2.1, Id(Bk)syl1 needs to be active in order to preserve full initial-syllable contrast among all vowels in the inventory. *F1's location between Id(Bk)syl1 and *F3, while not necessary (since *F3 could subsume the work that *F1 is doing), is also not problematic in and of itself. *F1B3 should not be active (since B3 contains /o/) but its position is not problematic as long as *F1 outranks it. However, the ranking values of *F1 and *F1B3 are just close enough that they switch places at evaluation every now and then. For example, whereas a grammar for North Estonian should consistently see input /ø..o/ producing a faithful output ø..o → ø..o, this grammar does so in 99% of test cases, with ø..o → ø..ø occurring the other 1% of the time. Tableaux TODO and TODO compare the two different ranking orders.

/ø..o/	Id(Bk)σ_1	*F1	*F1B3	*F3
a) ☞ ø..o		*	*	*
b) ø..ø		**!		**

Tableau TODO. The grammar chooses the intended winner when *F1 >> *F1B3.

/ø..o/	Id(Bk)σ_1	*F1B3	*F1	*F3
a) ø..o		*!	*	*

b) ☞ ø..ø			**	**

Tableau TODO. The grammar chooses the intended loser when *F1B3 >> *F1.

The same relationship results in a 99% success rate for the following inputs:

- ø..ø..ɯ → ø..o..i ~ ø..ø..i
- ø..o..ɯ → ø..o..i ~ ø..ø..i
- ø..o..ɑ → ø..o..ɑ ~ ø..ø..ɑ

As mentioned above, *F1's location between Id(Bk)syl1 and *F3 is not problematic in and of itself. But Id(Bk)syl1 and *F1 are just close enough that there is a small chance of them swapping places at evaluation. For example, whereas a grammar for North Estonian should consistently see input /ø..y..ɤ/ producing a faithful output ø..y..ɤ → ø..u..e, this grammar does so in 99% of test cases, with ø..y..ɤ → o..u..e occurring the other 1% of the time.

/ø..y..ɤ/	Id(Bk)σ_1	*F1	*F1B3	*F3
a) ☞ ø..u..e		*		*
b) o..u..e	*!			

Tableau TODO. The grammar chooses the intended winner when Id(Bk)σ_1 >> *F1.

/ø..y..ɤ/	*F1	Id(Bk)σ_1	*F1B3	*F3
a) ø..u..e	*!			*
b) ☞ o..u..e		*		

Tableau TODO. The grammar chooses the intended loser when *F1 >> Id(Bk)σ_1.

The same relationship results in a 99% success rate for the following inputs:

- ø..ɑ..o → ø..ɑ..o ~ o..ɑ..o

Overall, the results are nevertheless very good, with 267 of 270 learning forms and 827 of 830 testing forms producing correct outputs in 100% of samples.

4.2.2.2 - Finnish

These final ranking values produce a grammar that, interpreted categorically, aligns with the crucial rankings of *B2, *F3…B5, *F3B5, *B5…F3, *B5F3 >> Id(Bk)syl1 >> Id(Bk). And in fact, the relative distances between ranking values are great enough that the grammar produced correct outputs for all inputs (both learning and testing) in 100% of evaluations.

TODO sample tableaux?

4.3.3.3 - North Seto

These final ranking values produce a grammar that, interpreted categorically, meets the crucial target rankings of *F4…B5, *F4B5, *B5…F4, *B5F4, Id(Bk)syl1 >> *B1 >> Id(Bk). However, given the stochastic element in the evaluation stage, there are a few pairs of constraints whose relative ranking values are close enough so as to have resulted in the occasional swapped ordering, producing a small amount of (ungrammatical) variation.

First:

- *B1 103.934 >> *B1F5 96.66
- Although *B1F5 is satisfied by a large amount of the learning data (since only a Ii sequence would violate it), it should in fact be inactive due to the licit status of Ii sequences. It is relatively low compared to the other constraints involved in the crucial rankings for NSeto, but not quite low enough. *B1 (ok to be active) and *B1F5 (shouldn't be active because F5 contains i) are just close enough to swap places at evaluation every once in a while. So instead of Ii → Ii, in 1% of tests we got Ii → II instead. (/Ii/ → Ii 0.99 vs II 0.01)
- Todo tableau!
- Same issue:

Second:

- IdBkSyl1 113.6937 >> *B1 103.934
- IdBkSyl1 >> *B1 is correct; it's ensuring no I vowels in noninitial syllables. However, the two are just close enough to swap places at evaluation every once in a while. So instead of /IOa/ → Ioa, in 1% of tests we go /IOa/ → ioa instead. (/IOa/ → Ioa 0.99 vs ioa 0.01)
- Todo tableau!
- Same issue:

TODO data on frequency of successful results for various tableaux?

Testing data issues - 7 or TODO forms:

-
- /IiO/ → Iio 0.99 vs IIo 0.01
- /IiA/ → Iia 0.98 vs IIa 0.02
- /IiI/ → Iii 0.99 vs III 0.01
- /IIy/ → Iiu 0.98 vs IIu 0.02
- /IIe/ → IiE 0.99 vs IIE 0.01
- /IIo/ → Iio 0.99 vs IIo 0.01

Learning data issues - 3 of TODO forms:

-
- /Iiu/ → Iiu 0.98 vs IIu 0.01 vs iiu 0.01

- /Iou/ → Iou 0.99 vs iou 0.01

TODO. Methods for learning simulations

All learning simulations were run using my Python-based implementation of the GLA (see Appendix TODO). There are excellent existing implementations (TODO cite), but given the broad range of parameters and values I intended to test, I needed to write my own script to be able to fully customize the settings for my learners.

TODO.

REMOVE THIS SECTION. An ideal learner

TODO

0.99994 average frequency of good results
T_Mgen4.150.050fs_mg2_fs_sg30 (lowest num inputs above 90% good results = 1097)

- Without mg: 0.5019757575757573
- With mg1 instead: 0.8479090909090914
- With mg3 instead: 0.9982848484848493
- With mg4 instead: 0.9581242424242425
- Without fs: 0.9453818181818182
- With sg40 instead: 0.9996212121212124
- With sg20 instead: 0.9859909090909094
- With sg10 instead: 0.9457303030303029
- With sg0 instead: 0.9453727272727274
- Without sg: 0.9452303030303032
- With ReLU: 0.9996424242424243

4.2.1. Additional learning parameters

In this section I briefly introduce some of the learning parameters that were useful in improving the success of the learner on inputs from the sample languages. More detailed explanations, as well as test results involving variation of these parameters, are given in sections 4.3 through 4.5.

Promotion rate: how far should winner-preferring constraints get promoted at each learning update (as a fraction of the amount that loser-preferring constraints are demoted)?

F_{spec} >> F_{gen}: Faithfulness constraints that apply only in a specific context should be preferred over their more general counterparts.

- *A priori* bias: A specific faithfulness constraint should always outrank its general counterpart.
- Favour Specificity: In a case where either or both of the specific and general versions of a faithfulness constraint could be given credit for preferring the intended winner, only the specific one should receive it.

M_{gen} >> M_{spec}: More general markedness constraints should start with a higher initial value than more specific ones.

4.2.2. A successful learning simulation

There are a range of combinations of the parameters introduced in Section 4.2.1 that produce excellent learning results. In this section I present the results of a simulation run with one such combination of parameters. In sections 4.3, 4.4, and 4.5 I define these parameters in detail and contrast these results with those from learners with different values of the crucial parameters.

Parameter settings for "ideal" Learner A:

- Basic parameters set to default settings
- Promotion rate of type 2: 1 / *W*
- *A priori* faithfulness bias is set to 30
- Favour Specificity is on
- Markedness constraints are distributed by function $\mathscr{F}_4$, with m = 1.50 and b = 0.50.

Under these conditions, learning simulations for all three sample languages succeed magnificently, producing expected outcomes in 99.99% of test evaluations across the three languages. Test results are summarized in Table TODO.

Language	Average frequency of correct outputs
North Estonian	0.99995
Finnish	1.00000
North Seto	0.99988

Table TODO. Summary of results from learning simulations with Learner A.

4.2.2.1. North Estonian results

The final ranking values for a selection of crucial constraints, after learning from simulated North Estonian inputs, are shown in Table TODO. This grammar exhibits the crucial rankings of *B1 >> Id(Bk)syl1 >> *F3, *B2 >> Id(Bk) as presented in Section 3.TODO.

Constraint	Final ranking value
*B1	158.800
…	…
*B5…F3	150.000
*B5F3	150.000
*F5…B2	150.000
*F5B2	150.000
…	…
Id(Bk)syl1	115.938
…	…
*B2	103.271
…	…
*F3	72.392
…	…
Id(Bk)	52.471
…	…

Table TODO. Excerpt of final ranking values for North Estonian after learning simulation with Learner A.

4.2.2.2. Finnish results

Table TODO shows the final ranking values for a selection of crucial constraints, after learning from simulated Finnish data. This grammar aligns with the crucial rankings of *B2, *F3…B5, *F3B5, *B5…F3, *B5F3 >> Id(Bk)syl1 >> Id(Bk) as presented in Section 3.TODO.

Constraint	Final ranking value
*B2	168.000
…	…
*B5…F3	150.000
*B5F3	150.000
*F3…B5	150.000
*F3B5	150.000
…	…
Id(Bk)syl1	106.496
Id(Bk)	69.318
…	…

Table TODO. Excerpt of final ranking values for Finnish after learning simulation with Learner A.

4.2.2.3. North Seto results

Learning from simulated North Seto data results in final ranking values for a selection of crucial constraints shown in table TODO. This grammar meets the crucial target rankings of *F4…B5, *F4B5, *B5…F4, *B5F4, Id(Bk)syl1 >> *B1 >> Id(Bk) as presented in Section 3.TODO.

Constraint	Final ranking value
*B5…F4	150.182
*B5F4	150.182
…	…
*F4…B5	150.000
*F4B5	150.000
Id(Bk)syl1	113.694
*B1	103.934
…	…
Id(Bk)	64.761
…	…

Table TODO. Excerpt of final ranking values for North Seto after learning simulation with Learner A.