

Biases and Stages in Phonological Acquisition *

a dissertation defense by Anne-Michelle Tessier

1. Overview of the dissertation

The broad goals: bridging some gaps between learnability and acquisition

- bringing learnability proposals more in line with observed acquisition data
- testing the predictions of learnability proposals with experimental data

The theoretical goal: an error-driven learner that is both restrictive and gradual

- a learner's grammar is maximally *restrictive* if it generates all the observed outputs but as few others as possible
- a learner is *gradual* if it persists in making the same errors for some time despite evidence, and if it adopts only *some* of the target rankings at each new stage.

1) What's in an error?

observed, target English output: 'toast!' [tost] (says mother)
current grammar's output: 'to!' [to] (says child)

(a) an error tableau:

/tost/	NoCoda	*ComplexCoda	*Pharyngeal	Max-C
tost	*!	*!		
☞ to				*

(b) comparison of the two forms:¹

winner ~ loser	NoCoda	*ComplexCoda	*Pharyngeal	Max-C
tost ~ to	L	L	e	W

* Here is the Coles Notes version of my acknowledgements: Special thanks to Joe Pater and John McCarthy; to Shelley Velleman and Lyn Frazier; to Kathryn Flack, Michael Becker, Karen Jesney, Tim Beechey, Della Chambless, John Kingston, Bruce Hayes, Alan Prince and Bruce Tesar; to audiences at BUCLD30, MOT2006, the 2006 LSA meeting, the Brown/Umass 2006 Workshop and WCCFL25; and to Marcin Morzycki, Paula Menéndez-Benito, Ana Arregui, Jan Anderssen and Kyle Johnson. Thanks also to you, for showing up even though it's summer and far too humid and so forth.

¹ This comparison format is taken from Prince (2002) – there, such things are called 'Elementary Ranking Condition vectors', and in this dissertation I refer to them as ERC rows.

The issue: what should the learner learn from (1b)?

- We know that English learners eventually learn that codas and complex codas are grammatical, but that pharyngeals aren't
- ... and more generally, that learners acquire the generalizations of their language without getting evidence of what isn't allowed (e.g. no pharyngeals)

Upshot: they must be learning restrictively.

- We also know that along the way to learning to produce complex codas, many English learners stop off for some months at an intermediate stage where they produce only singleton codas:

2) A common trajectory of coda acquisition

initial state: /tost/ → [to]
intermediate state: /tost/ → [tos]
final state: /tost/ → [tost]

- ... and more generally, that learners take years to get from initial to final grammars – even when they have the evidence of words like 'toast'

Upshot: they must be learning gradually.

Why restrictive learning and gradual learning are tricky to reconcile:

To be restrictive:

- a learner must be efficient at learning from the evidence of its errors and extracting *all* the ranking arguments from those errors
- to learn the right final ranking, the learner must notice from (1b) that *NoCoda is violated by target English words like "toast", but *Pharyngeal isn't

But to be gradual:

- a learner can't learn from all of its evidence right away
- to go through the intermediate [tos] stage of (2), a learner must temporarily ignore the fact that *ComplexCoda is also violated by English words like "toast"

1.1 The main theoretical proposal: Error-Selective Learning

- an existing efficient method of building rankings: *Biased Constraint Demotion* (Prince and Tesar, 2004: see also Hayes, 2004), combined with
- a new way of choosing errors to learn from: the *Error-Selection* algorithm

- BCD: powerful way of finding restrictive grammars consistent with a set of errors
- Error Selection: a gradual way of feeding BCD with errors that inch it toward the target grammar

At the core of the proposal: the Support, a memory of errors

- Prince and Tesar (2004): the Learning Support
- the Support is the learner's sum total of ERC rows, like the one in (1b), from all of their errors to date

The Support is where the learning action is – not in the rankings

- the BCD algorithm is just a function that takes a Support and returns a grammar
- the rankings it builds are only as right as the data it has, and they can be changed at any time
- on the other hand, the Support is built gradually and continuously updated
- it is the language-specific, on-going record of why the learner has the ranking it does, and what evidence there is for every part of it

The Support is crucial to learning all kinds of language-specific things in learning (back to this in section 5 of the talk)

In my proposal:

- the Error Selection algorithm uses its knowledge of BCD to decide what errors to commit to the Support

Also:

- I argue that the use of stored errors is the main advantage of error-selective BCD over any approach like the Gradual Learning Algorithm (Boersma, 1997)

1.2 Overview of the chapters

1. Preliminaries: the Biased Constraint Demotion algorithm I use

- what are the biases, and why?
- in particular: how to impose a bias for more specific faith constraints
- the importance of the Support in imposing persistent biases

2. The proposal: Error-Selective learning

- data: the kinds of stages I want my learner to go through
- theory: the way ESL gets errors into the Support
- discussion of ESL, including the connections to frequency and variation

3. The alternative in the literature: the Gradual Learning Algorithm

- how the GLA works
- why it needs all the same ranking biases
- how its lack of stored errors prevent it from imposing persistent biases

4. An experimental test of biases and stages

- an artificial language wug-test with 4-year olds
- designed to test for one proposed ranking bias: undominated OO-faith
- also provides evidence of the kind of intermediate stages, influenced by OO-faith, that my Error-Selective learning can derive

1.3 Overview of the talk

- section 2: summarizes error-driven BCD learning and chapter 1
- sections 3-4: presents the Error-Selective learning proposal of chapter 2
- section 5: highlights the role of errors in my proposal, as compared to the alternative of the GLA, as discussed in chapter 3

2. The essentials of chapter 1

2.1 Error-Driven Learning using a Constraint Demotion algorithm

(see especially Tesar and Smolensky, 2000)

The BCD learner proceeds in the following error-driven way:

- The learner has a current ranking
- It observes some output form of the target language
e.g. an adult says “toast”!
- It feeds that output to its current ranking to see whether the current grammar can reproduce that output form faithfully
e.g. does the input ‘toast’ come out as [tost] or [to]?

3) the error-creating tableau from (1a):

/tost/	NoCoda	*ComplexCoda	*Pharyngeal	Max-C
tost	*!	*!		
☞ to				*

- If the current grammar maps the output form onto something else, the learner notices this mapping as an error
and e.g. builds an ERC row for tost ~ to

4) the ERC row repeated from (1b)

winner ~ loser	NoCoda	*ComplexCoda	*Pharyngeal	Max-C
tost ~ to	L	L	e	W

- When such an ERC row is added to the Support, the BCD algorithm builds a new ranking that will choose all the Support’s winners over its losers

e.g. build a ranking that considers /tost/ → [tost]
more optimal than /tost/ → [to]

How to ensure that [tost] is more optimal, informally:

by installing constraints that are better satisfied by ☞ /tost/ → [tost]
above constraints that are better satisfied by * /tost/ → [to]

How to ensure that [tost] is more optimal, a little more formally:

by installing *some* W-preferring constraint
above *ever* L-preferring constraint²

- After applying BCD, the learner has a new ranking:
e.g.: Max-C >> NoCoda, *ComplexCoda

- ... and this ranking no longer makes any of the errors in the Support

5) the new tableau for ‘toast’, which no longer causes errors

/tost/	Max-C	NoCoda	*ComplexCoda
☞ tost		*!	*!
to	*		

2.2 The need for *Biased* Constraint Demotion

The main challenge for the error-driven learner of OT

- lots of different rankings are consistent with the same Support
- how should it choose the most restrictive?

2.2.1 What’s the danger of not learning restrictively?

- an example: the French stress rule, which is ‘stress the final syllable’

6a) an error the French learner could make:³

/papá/	Trochee	lamb	Faith-Stress
(pa.pá)	*		
☞ (pá.pa)		*	*

² This is the Cancellation/Domination Lemma of Prince and Smolensky (1993); as rephrased a bit in Prince and Tesar (2004): 255.

³ To make the example as simple as possible, this tableau has just three constraints – which means that to make the error, the learner must have a ranking of Trochee >> lamb. In chapter 1, I give this example using a version of CON that includes Non-Finality as well. In that case, any grammar that simply ranks all M >> all F will produce prefer bisyllabic words with initial over final stress, and so produce this error.

6)b the resulting ERC row in the Support

<i>winner ~ loser</i>	Trochee	lamb	Faith-Stress
(papá) ~ (pápa)	L	W	W

What ranking does the new grammar need to prefer papá over pápa?

either lamb >> Trochee
or Faith-Stress >> Trochee

If the learner chooses the latter, its new ranking is right in that it produces the right stress in [papá]:

7)a Faith-Stress >> Trochee resolves the error...

/papá/	Faith-Stress	Trochee	lamb
☞ (pa.pá)		*	
(pá.pa)	*!		*

But more generally, this grammar is wrong in that its stress is *lexically* determined – it just falls wherever it is in the input:

7)b) ... but Faith-Stress >> Trochee would make other errors

/dádý/	Faith-Stress	Trochee	lamb
(da.dý)	*!	*	
☞ (dá.dy)			*

The generative assumption: (7) is NOT the grammar of French

- it is not just an accident of the lexicon that every French word has final stress
- it is also a fact we want to attribute to (and capture in) the phonological grammar of French speakers

The learnability issue: how to realize that (7) is the wrong French grammar?

- this grammar isn't French, but it won't prompt any more errors
- so if errors are what drives learning: the learner will never give up this grammar
- this ranking is therefore a superset 'trap'

2.2.2 How to learn restrictively?

The well-known OT learnability response to the problem above:

- learners must prefer rankings of Markedness >> Faithfulness⁴
- more generally: the heuristic of the Subset principle (Berwick, 1985)

⁴ although cf. Bernhardt and Stemberger (1998) and Hale and Reiss (1998).

an incomplete history of subsets and language learnability: Angluin, 1980; Berwick, 1985; Dresher and Kaye, 1990; Gibson and Wexler, 1994; Dresher, 1999; in OT: Smolensky, 1996; McCarthy, 1998; Pulleyblank and Turkel, 1998; Ito and Mester, 1999; Smith, 2000; Prince and Tesar, 2004; Hayes, 2004

2.3 How BCD uses ranking biases

not: take an existing grammar and change some of its rankings

instead: start with the data in the Support and *build a new ranking from scratch*

How BCD goes from errors to rankings

- the BCD learner builds the ranking from top to bottom: first stratum to last
- in building constraint strata, the learner's goal is still to install constraints that will prefer the winning outputs it has encountered – [tost] -- over the errors that previous grammars have made – [to]
- and its version of ranking biases is *the order in which it attempts to install constraints* in each stratum

For the 'toast' example we saw, the M >> F bias is all we need:

8) a Support with two ERC rows:

Winner ~	NoCoda	*ComplexCoda	*Pharyngeal	Max
tost ~ to	L	L	e	W
piz ~ pi	L	e	e	W

9) The BCD reasoning

a) first, install all the Markedness constraints *that don't prefer any losers*:
resulting stratum 1: *Pharyngeal

b) again, install all the M constraints *that don't prefer any losers*:
... but both NoCoda and *ComplexCoda do prefer losers
... i.e., they were responsible for making these errors...

c) so next, try installing a Faithfulness constraint *that prefers a winner*:
... there's only one F constraint here, Max, so:
resulting stratum 2: Max

d) and already our new grammar will no longer make the errors in 8)...
... so we can install all the remaining M constraints and we're done:
resulting stratum 3: *NoCoda, *ComplexCoda

e) the final result: the right, restrictive end-state grammar of English.
*Pharyngeal >> Max >> NoCoda, *CompCoda

2.3.1 About the full set of ranking biases

Markedness >> Faithfulness is not the whole story.

The full ordering in the version of BCD that I use is in 10) below:

10) OO-Faith >> Markedness >> Specific-IO-Faith >> General-IO-Faith

The reasoning behind OO-Faith >> everything else:

- for the restrictiveness argument, see McCarthy (1998); Hayes (2004), and see the implementation in Becker (2006); for anecdotal acquisition evidence, see Kazazis (1969), Smith (1973), discussed by Hayes (2004), Jesney (2005) respectively

The reasoning behind Specific >> General IO-Faith

- for the restrictiveness argument, see Smith (2000), Hayes (2004)
- in chapter 2 I provide data from the literature and from the Compton/Streeter/Pater database (Compton and Streeter, 1997) of a variety of intermediate stages that require such a ranking; see also Revithiadou and Tzakosta, 2004

About the Specific >> General IO-Faith bias:

- we will see that this bias is crucial to how Error-Selective learning works
- however: Prince and Tesar (2004) explicitly reject such a bias, and use different principles to determine which IO-faith constraints to install in any stratum.

My approach to the Specific >> General IO-faith bias (chapter 1, §4-6)

- calculate the specificity of faithfulness constraints, when necessary, by looking directly at the Support's winners (the observed forms of the language)
- see also Hayes (2004) for a version of this proposal

2.4 Interim Summary

The BCD algorithm learns restrictive grammars, by

- storing each error in the Support
- analyzing exactly which rankings are required to stop making each error

The consequence of this Support-driven learning:

- errors reveal all, immediately
- as soon as an error gets into the Support, BCD will find a ranking that explains it
- in other words: BCD has no way to learn *partially* from an error

The rest of the talk: using this BCD algorithm to learn gradually

3. The data from Chapter 2: Intermediate Stages of Acquisition

- first: characterizing two kinds of intermediate stage data
- then the Error-Selective approach

3.1 Coda clusters in English and Dutch

Children acquiring languages with complex coda structures (like English and Dutch) often go through the intermediate stage discussed in (2):

- 11) /CVC/ → [CVC], *[CV]
/CVCC/ → [CVC], *[CVCC]

- 12) *Children at an intermediate stage of coda acquisition*
(see esp. Levelt, Schiller and Levelt, 1994; Levelt and van der Vijver, 2004)

	Singleton Codas: retained			Complex Codas: reduced to singletons		
	Target	Child	Gloss	Target	Child	Gloss
<i>Dutch:</i> <i>Eva</i> (1;4,12) (Fikkert, 1994; Levelt, 1994)	/te:n/	[teɪ]	'toe'	/e:nt/	[eɪ]	'duck'
	/bɛd/	[dɛ]	'bed'	/stɑ:rt/	[tɑ]	'tail'
<i>English:</i> <i>G</i> (2;3-2;9) (Gnanadesikan 1995/2004)	/grɛ:p/	[gɛp]	'grape'	/drɪŋk/	[brɪk]	'drink'
	/pɪz/	[pɪz]	'peas'	/frɛnd/	[fɛn]	'friend'
	Singleton Codas: retained or deleted			Complex Codas: reduced or deleted		
<i>English:</i> <i>PJ</i> (1;11) (Demuth and Fee, 1995; Demuth, 1996)	/wɑ:k/	[rɑʔ]	'walk'	/tɔ:st/	[tɔ:s]	'toast'
	/sʊp/	[sʊp]	'soup'	/bɪdz/	[bɪ:s]	'beads'
		[su:], [su]			[be:]	
/dʒʊ:s/	[dʒu:s], [du:s]	'juice' ⁵				
	[dʒu:]					

The ranking for this intermediate stage:

- 13) NoComplexCoda >> Max >> NoCoda

⁵ In Demuth (1996), the child *and* adult targets are transcribed as "dz", rather than "dʒ". I assume this is a typo, but in any event the quality of the child's onset is not important here.

14)a) Max >> NoCoda protects singleton codas

/gre ⁱ p/	Max	NoCoda
☞ [ge ⁱ p]		*
[ge]	*!	

14)b) *ComplexCoda >> Max reduces coda clusters

/frend/	NoComplexCoda	Max	NoCoda
[fend]	*!		*
☞ [fen]		*	*
[fe]		*!*	

More generally: this is a stage at which *some* Markedness has been demoted, but not enough to get to the target grammar.⁶

3.2 Onset clusters in Quebecois French

From Rose (2000): an intermediate stage in the acquisition of Quebecois French where complex onsets are preserved faithfully in *stressed syllables* only:

- 15) /CV.ⁱCCV/ → [CV.ⁱCCV], *[CV.ⁱCV]
/CCV.ⁱCV / → [CV.ⁱCV], *[CCV.ⁱCV]

16) Clara and Théo at the intermediate stage of onset acquisition

	Stressed Syllables: complex onsets retained			Unstressed Syllables: complex onsets reduced		
	Target	Child	Gloss	Target	Child	Gloss
Clara 1;09.29- 2;03.05	/bi. ⁱ bɛʒ/	[pa. ⁱ pɛʒ]	baby bottle	/fʁi. ⁱ go/	[bu. ⁱ ko]	fridge
	/gʁis/	[kʁis]	slides	/bɛy. ⁱ le/	[bɛ. ⁱ le]	burned
	/si. ⁱ tɥuj/	[θə. ⁱ tɥuj]	pumpkin	/gʁi. ⁱ sad/	[ka. ⁱ sæd]	slide
	/pʁœs/	[pʁœɣ]	cries	/tɥu. ⁱ ve/	[tu. ⁱ ve]	found
Théo 2;05.29- 2;11.29	/gʁo/	[gʁo]	big	/tɥak.tœs/	[ta ⁱ tœ ^u]	tractor
	/tɥɛ̃/	[kɥɛ̃]	train	/gʁy.jo/	[k ^h œ. ⁱ jo]	oatmeal
	/kle/	[kxi]	key	/tɥu.ve/	[ku. ⁱ βi]	found
	/pʁœs/	[pʁœ ^u]	cries	/kɥɛm.gʁa. ⁱ se/	[kɥa ⁱ ..na. ⁱ se]	ice cream

⁶ That these two M constraints are in a subset relation turns out not to be crucial: ch.2 §4.4.3

The ranking for this intermediate stage:

- 17) Max(Stressed-σ) >> NoComplexOnset >> Max

18)a) *ComplexOnset generally reduces onsets to singletons:

/bɛy. ⁱ le/	*ComplexOnset	Max
☞ [bɛɪ. ⁱ le]		*
[bɛ. ⁱ le]	*!	

18)b) Max-(Stressed-σ) >> *CompOns protects them just when stressed

/pʁœs/	Max(Stressed-σ)	*ComplexOnset	Max
☞ [pʁœ ^u]		*	
[pʁœ ^u]	*!		*

More generally, this stage can be schematized as:

- 19) Specific-Faith >> Markedness >> General-Faith

3.3 Summarizing some more intermediate stages

Intermediate M stages like (14)

- unmarked complex onsets retained, but marked ones reduced (ch.2 §2.2.2)
- the 'one foot' stage – syllables that get parsed into the output head foot retained, but others truncated (see e.g. Fikkert, 1994; Curtin and Zuraw, 2001)

Intermediate F stages like (19)

- English: initial stressed syllables retained, but initial unstressed ones deleted (Kehoe, 2000)
- Spanish, Catalan, Portuguese: codas are retained in stressed syllables, but deleted in unstressed syllables (e.g. Lléo, 2003)
- Greek: initial unstressed syllables are retained, but non-initial unstressed syllables are deleted (Revithiadou and Tzakosta, 2004)

3.4 The connections between BCD and these stages

1. Adding just one error with an English complex coda or a French unstressed complex onset to the Support will get BCD to the right end-state grammar, steamrolling past the intermediate stages. This is the excessive efficiency of BCD.
2. These intermediate stages stick closer to the ranking biases than the final state. That is – the BCD learner would happily build the intermediate stage ranking if only the Support didn't contain some of its particularly marked errors.

So: the Error-Selective learner will ensure that the Support *doesn't* contain those particularly marked errors, for a while.

4. The proposal from chapter 2: Error-Selective Learning

4.1 What Happens when the Error-Selective BCD learner makes an error

- errors don't immediately appear in the Learning Support once they're made... nor do they trigger re-ranking
- instead, errors are filed in a temporary storage area: the **Error Cache**

20) At an Early Stage: this error is made...

/tost/	NoCoda	*CompCoda	Max-Seg
tost	*!	*	
tos	*!		*
to			**

adding error to the Cache

21)a) ... the learner adds it to the current Error Cache:

Input	Winner ~Loser	NoCoda	*CompCoda	Max-Seg
/tost/	tost ~ to	L	L	W
/piz/	piz ~ pi	L	e	W

21)b) ... but the Learning Support Table is NOT updated:

Input	Winner ~ Loser	NoCoda	*CompCoda	Max-Seg
... empty, waiting...				

... and nothing else happens. More errors are made, and the Cache grows...

4.2 What drives the Error-Selective learner to learn

- Re-ranking is triggered when a constraint overcomes the **Violation Threshold**
- that is: when some constraint has assigned Ls to more than x words in the Error Cache
 - we'll call that particularly offending constraint the **Trigger Constraint**: because it has triggered learning

If e.g. the violation threshold is 3, then as soon as some constraint assign an L to 3 different winner ~ loser pairs in the Error Cache, learning is triggered, e.g.:

22) a (sample of an) Error Cache that triggers learning:

Input	Winner ~Loser	NoCoda	*CompCoda	Max	*CompOnset
i) /tost/	tost ~ to	L	L	W	L
ii) /piz/	piz ~ pi	L	e	W	e
iii) /gre'p/	gre'p ~ ge	L	L	W	L
iv) /ti/	ti ~ si	e	e	e	e

4.3 Step 1 of Error-Selective Learning: Choosing the Best Error

23) The Error Selection Algorithm (ESA)

- Choose as the best error that row in the Cache which:
- a) has an L assigned by the Trigger Constraint, and of those
 - b) has the fewest Ls assigned by other Markedness constraints, and of those
 - c) has the most Ws assigned by Faithfulness constraints

Looking at 22), we only need the first two criteria to choose the best error:

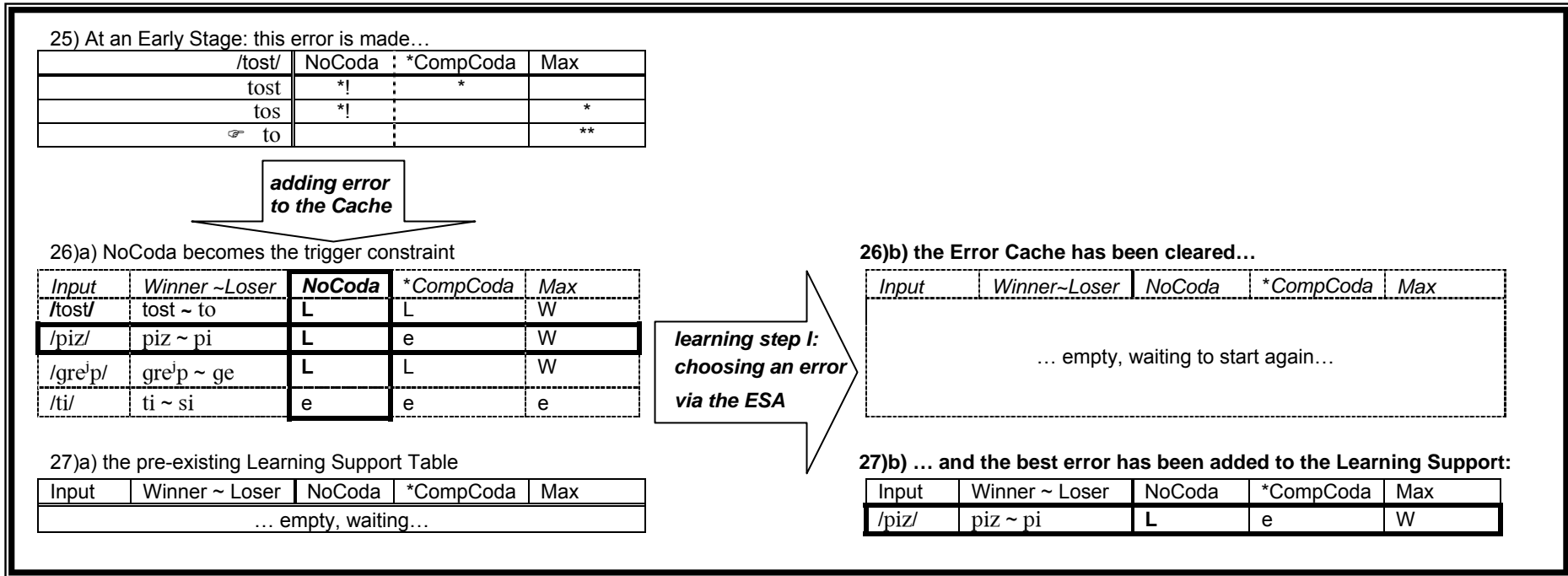
- criterion (a) eliminates 'tea', because it does not have a NoCoda violation,
- criterion (b) eliminates 'toast' and 'grape', because they also have Ls assigned by *ComplexCoda and *ComplexOnset so:

24) piz ~ pi is the chosen Best Error

Input	Winner ~Loser	NoCoda	*CompCoda	Max	*CompOnset
i) /tost/	tost ~ to	L	L	W	L
ii) /piz/	piz ~ pi	L	e	W	e
iii) /gre'p/	gre'p ~ ge	L	L	W	L
iv) /ti/	ti ~ si	e	e	W	e

Once the best error has been chosen, the Error Cache gets cleared, and the error gets added to the Support. Turn over to see this:

4.4 How Error-Selective BCD Learner gets from errors to a new Support



Why does the ESA work this way?

- criterion (a), and the idea of a Violation Threshold, make the learner attend to the most frequent Markedness problem
- criterion (b) makes the learner learn *as little else as possible*⁷

4.5 Step 2 of Error-Selective Learning: Using BCD

Now that the learner has chosen a best error:

- the Support in (27b) has been updated
- this means the current ranking is now outdated, because it was built via BCD from a Support that didn't contain this latest error, piz ~ pi

So: any update of the Support prompts the Error-Selective learner to a fresh application of BCD

28) Using BCD to build a ranking from the Support in (27b):

- first, install all the Markedness constraints *that don't prefer any losers*:
resulting stratum 1: *ComplexCoda
- again, install all the M constraints *that don't prefer any losers*:
... but NoCoda does prefer a loser...
- so next, try installing a Faithfulness constraint *that prefers a winner*:
... there's only one F constraint here, Max, so:
resulting stratum 2: Max
- and already our new grammar will no longer make the errors in 27b)...
... so we can install all the remaining M constraints and we're done:
resulting stratum 3: *NoCoda
- the final result: the right, restrictive end-state grammar of English.
*ComplexCoda >> Max >> NoCoda

⁷ and see chapter 2 for the reasoning behind criterion (c)

4.6 How Error-Selective BCD Learner gets from errors to a new ranking – the final flow chart

29) At an Early Stage: this error is made...

/tost/	NoCoda	*CompCoda	Max
tost	*!	*	
tos	*!		*
☞ to			**

adding error
to the Cache

30)a) NoCoda becomes the trigger constraint

Input	Winner ~Loser	NoCoda	*CompCoda	Max
/tost/	tost ~ to	L	L	W
/piz/	piz ~ pi	L	e	W
/gre'p/	gre'p ~ ge	L	L	W
/ti/	ti ~ si	e	e	e

learning step I:
choosing an error
via the ESA

30)b) the Error Cache has been cleared...

Input	Winner~Loser	NoCoda	*CompCoda	Max
... empty, waiting to start again...				

31)a) the pre-existing Learning Support

Input	Winner ~ Loser	NoCoda	*CompCoda	Max
... empty, waiting...				

31)b) ... and the best error has been added to the Learning Support:

Input	Winner ~ Loser	NoCoda	*CompCoda	Max
/piz/	piz ~ pi	L	e	/piz/

learning step II:
re-ranking
via BCD

32)a) The Next Stage: is *Intermediate*. It maps the Support's error faithfully:

/piz/	*CompCoda	Max	NoCoda
☞ piz			*
pi		*!	

32)b) ... but it makes a new, *partially faithful* error on 'friend':

/tost/	*CompCoda	Max	NoCoda
tost	*!		*
☞ tos		*	*
to		**!	

... And with respect to coda clusters, the grammar in 32) is the intermediate stage from (2)!

4.7 At the end-state: Error-Selective learning is equivalent to BCD learning

Error-Selective learner gradually converges on the final grammar that BCD would have found anyway. Just slower.

Why?

- every time a constraint exceeds the Violation Threshold, some new error is chosen to add to the Support
- once BCD learns from an error -- it is never made again
- because the Cache is emptied at the end of learning -- the next cycle of learning will teach the learner something new about the target
- eventually all the errors necessary to finding the correct grammar will be added to the Support – and this is the final state

4.8 The connection between ESL and Frequency

The role of frequency: mediated through VTs.

With respect to Trigger Constraints:

- because learning is triggered when a sufficient numbers of losers have violated the same constraint, *more general M constraints trigger learning before more specific ones* (compare NoCoda and *ComplexCoda)

With respect to order of acquisition of marked structures:

- broadly speaking, markedness constraints are demoted in accordance with how frequently they are violated in the target
- this is the right kind of empirical result (see, e.g. Roark and Demuth, 2000; Kirk and Demuth, 2003; Stites, Demuth and Kirk, 2004)

5. The Importance of Errors

5.1 A memory for errors: the roles for of the Support

Having a Support makes Error-Selective learning possible

- updating the Support is what triggers learning
- so delaying the update of the Support, using the Cache and Violation Thresholds, is what lets the Error-Selective learner delay learning

- the Support is also crucial to the suggestion I make in chapter 2 about how Error-Selective learning could be extended to model variation, or vacillation between intermediate stages⁸

Beyond Error Selection, the Support is crucial to the BCD learner in many ways

The Support allows the learner to compare across and between errors

- to work out hidden structure like footing (Tesar, 2000)
- to diagnose exceptionality, as compared to variation, and determine the necessary lexically-specific constraints (Pater, to appear)
- to determine whether faithfulness constraints are in language-specific subset relations, so that the Specific-Faith >> General-Faith ranking bias can correctly be enforced (Tessier, 2006, and see chapter 1)

The Support allows the BCD learner to escape superset grammars

The central genius of the BCD approach⁹ is that it only saves old *errors*, and not old *rankings*

- if the Support's data is currently incomplete or even wrong, the learner may temporarily choose a superset grammar
- but as soon as the Support changes, BCD will find the new most restrictive ranking it can
- because every grammar is built from scratch – any ranking can be changed, and any ranking for which there is no longer any evidence will be abandoned.

⁸ Even the variable ESL approach I propose clearly does not handle all attested kinds of variation between stages that we see in learning. Two important cases it cannot explain: (i) U-shaped learning, (ii) fossilized forms, which might be described as variation between older and newer grammars. These phenomena need to be handled with other mechanisms or assumptions. Thanks especially to Shelley Velleman for discussion.

⁹ I am allowed to say this because BCD is not my invention...

- Dresher, B. Elan. (1999). "Charting the Learning Path: Cues to Parameter Setting," *Linguistic Inquiry* 30.2: 27-67. Fikkert, Paula (1994). *On the Acquisition of Prosodic Structure*. Ph.D. dissertation, University of Leiden.
- Gibson, Edward and Kenneth Wexler (1994). "Triggers". *Linguistic Inquiry* 25.3: p.407-454.
- Gnanadesikan, Amahlia (1995/2004). Markedness and Faithfulness Constraints in Child Phonology. In Kager, Rene, Joe Pater & Wim Zonneveld, (eds.), *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge, U.K.: Cambridge University Press.
- Goldwater, Sharon and Mark Johnson (2003). *Learning OT Constraint Rankings Using a Maximum Entropy Model*. Proceedings of the Workshop on Variation within Optimality Theory, Stockholm University.
- Hale, M. and C. Reiss (1998). "Formal and empirical arguments concerning phonological acquisition," *Linguistic Inquiry* 29: 656-683.
- Hayes, Bruce (2004). Phonological Acquisition in Optimality Theory: the early stages. In Kager, Rene, Joe Pater & Wim Zonneveld, (eds.), *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge, U.K.: Cambridge University Press.
- Ito, Junko and Armin Mester (1999). "The structure of the Phonological Lexicon" . In N. Tsujimura, (ed.), *The Handbook of Japanese Linguistics*. Oxford: Blackwell. pp. 62-100.
- Jesney, Karen (2005). *Chain shift in Phonological Acquisition*. M.A. thesis, University of Calgary.
- Kazazis, Kosta (1969). "Possible evidence for (near)-underlying forms in the speech of a child," in R. Binnick et al, eds., *Proceedings of the Chicago Linguistic Society* 5, Chicago, CLS.
- Kehoe, Margaret (2000). "Truncation Without Shape Constraints: The Latter Stages of Prosodic Acquisition," *Language Acquisition* 8.1: 23-67.
- Kehoe, Margaret and Geraldine Hilaire-Debove (2004). The Structure of Branching Onsets and Rising Diphthongs: Evidence from the Acquisition of French. In Alejna Brugos, Linnea Micciulla, and Christine E. Smith (eds.), *Proceedings of BUCLD 28*. pp. 282-293. Somerville, MA: Cascadilla Press.
- Levelt, Claartje C., Niels O. Schiller, and Willem J.M. Levelt (1999). "A developmental grammar for syllable structure in the production of child language." *Brain and Language* 68: 291-299.
- Levelt, C. C. and R. van de Vijver (2004) "Syllable types in cross-linguistic and developmental grammars," in R. Kager, W. Zonneveld, J. Pater, eds., *Fixing Priorities: Constraints in Phonological Acquisition*, Cambridge, Cambridge University Press.
- Lléo, Conxita (2003). Prosodic licensing of codas in the acquisition of Spanish. *Probus* vol. 15. pp 257-281.
- McCarthy, John J. (1998) "Morpheme structure constraints and paradigm occultation," in M. Catherine Gruber, Derrick Higgins, Kenneth Olson, and Tamra Wysocki, eds., *Proceedings of the Chicago Linguistic Society* 5, Vol. II: *The Panels*, Chicago, CLS.
- Pater, Joe (to appear). "The Locus of Exceptionality: Morpheme-Specific Phonology as Constraint Indexation". To appear in L. Bateman, M. O'Keefe, E. Reilly and A. Werle (eds.) *UMOP32: Papers in Optimality Theory III*. Amherst: GLSA.
- Prince, Alan (2002b). "Entailed Ranking Arguments". Manuscript, Rutgers University. ROA-500. 117pp
- Prince, Alan and Bruce Tesar (2004). Learning Phonotactic Distributions. In Kager, Rene, Joe Pater & Wim Zonneveld, (eds.), *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge, U.K.: Cambridge University Press.
- Pulleyblank, Douglas & William J. Turkel (1998). "The logical problem of language acquisition in optimality theory," in P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis & D. Pesetsky, eds., *Is the best good enough? Optimality and competition in syntax*. Cambridge, MA: MIT Press, 399-420.
- Revtithiadou, A. and M. Tzakosta (2004) "Markedness Hierarchies vs. Positional Faithfulness and the Role of Multiple Grammars in the Acquisition of Greek," *Proceedings of Generative Approaches to Language Acquisition (GALA) 2003*, Utrecht, LOT Occasional Series.
- Roark, Brian and Katherine Demuth (2000). Prosodic Constraints and the Learner's Environment: a Corpus Study. In S. C. Howell, S. A. Fish, and T. Keith-Lucas (eds.), *Proceedings of BUCLD 24*. Somerville, MA: Cascadilla Press. pp. 597-608.
- Rose, Yvan (2000). Headedness and Prosodic Licensing in the L1 Acquisition of Phonology. Ph.D. dissertation, McGill University.
- Smith, Jennifer L. (2000) "Positional faithfulness and learnability in Optimality Theory," in Rebecca Daly and A. Rehl, eds., *Proceedings of ESCOL99*, Ithaca, CLC Publications.
- Smith, Neilson V. (1973). *The Acquisition of Phonology: A Case Study*. Cambridge, MA: Cambridge University Press.
- Smolensky, Paul (1996) "On the comprehension/production dilemma in child language," *Linguistic Inquiry* 27: 720-731.
- Stites, Jessica, Katherine Demuth and Cecilia Kirk (2004). Markedness versus frequency effects in coda acquisition. In Alejna Brugos, Linnea Micciulla, & Christine E. Smith (eds.), *Proceedings of BUCLD 28*. Somerville, MA: Cascadilla Press. 565-576.
- Tesar, Bruce (2000). "Using inconsistency detection to overcome structural ambiguity in language learning." Technical Report RuCCS-TR-58, Rutgers Center for Cognitive Science, Rutgers University. ROA-426.
- Tesar, Bruce, John Alderete, Graham Horwood, Nazarré Merchant, Koichi Nishitani and Alan Prince (2003). "Surgery in language learning." In G. Garding and M. Tsujimura (eds), *Proceedings of WCCFL 22*. Somerville, MA: Cascadilla Press. pp. 477-490.
- Tesar, Bruce and Paul Smolensky (2000). *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.
- Tessier, Anne-Michelle (2006). "Learning Stringency Relations and the Structure of Faithfulness". 80th LSA meeting, Albuquerque, New Mexico.