CHAPTER III

ERROR-SELECTIVE LEARNING

1. Introduction

Chapter 2 presented the case for OT learning via a Biased Constraint Demotion-style algorithm. While I have proposed some augmentation of Prince and Tesar (2004)'s original proposals with additional biases and associated calculations, the approach is still very much committed to Prince and Smolensky's Cancellation/Domination Lemma (chapter 2 ex. 3) as the method of learning. In other words, getting from the current grammar to the next always involves reasoning from a set of ERC rows to the rankings that will choose winners over losers.

Biased Constraint Demotion is a mechanism for learning *everything* necessary to find the right rankings; as we will now see, this means it is not designed in any way to learn gradually or imperfectly. As we just saw at the end of chapter 2, gradual, realistic human learning is *not* Prince and Tesar's goal – they carefully point out that their aim is a formal OT ranking algorithm that behaves in accordance with the subset principle, and not one that acts like a human child.

However: the issues of restrictiveness discussed in chapter 1 are ones that should not be ignored in the investigation of early child phonologies (recall the discussion of French stress from the beginning of chapter 2 §3.) To reach their target phonological grammar, real-life language learners must find a way to replicate the observed outputs that they hear while still being restrictive. To the extent that BCD is the best way we

know of keeping an OT learner restrictive, it therefore seems worthwhile to examine how a BCD learner could also be as gradual as a human child.

The goal of this chapter is therefore to propose a novel way of combining BCD with something that derives stages of acquisition – a way that relies on the insights of constraint demotion and ranking biases to derive some broadly-attested developmental stages in L1 learning. The proposal is called Error-Selective Learning: a method to make the learner choosy about *which* errors it adds to the Support, and therefore uses to re-rank via BCD.

This introductory section sketches Biased Constraint Demotion's over-quick learning, foreshadows the Error-Selective proposal, and provides two key examples of the attested intermediate acquisition stages that Error-Selective learning can derive. At the end of this section I provide a roadmap to the entire chapter, so that readers more interested in either data or theory can decide where to focus and where to skim.

1.1 The approach to reconciling BCD and gradual learning

The pure BCD learner performs its total re-ranking every time an error is added to the Support, which is every time an error is made. Since BCD is so efficient, it will learn everything there is to learn from each error as it is made, and ensure such an error is never made again.

Thus, one major way in which BCD is not a model of real-time language learning is that it fails to go through any intermediate stages: that is, it cannot learn *partially* from any of its data. For example, a learner whose current grammar does not permit any codas

to surface (described by the ranking fragment in (1)a below), only needs to make an error on a single word with a complex coda to acquire the fully-correct ranking in 1b):

1) *Starting state*: NoCoda, *ComplexCoda >> Max
Target state: Max >> NoCoda, *ComplexCoda

A representative flow chart of this process, using the English word 'toast', is sketched below:

2) *How the BCD learner gets from an error to a new grammar*

a) *At an Early Stage: this error is made*

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

adding error to the Support

b) *... the error is put into the Learning Support Table…*

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

learning: re-ranking via BCD

c) *... and the next stage is the Final Stage, with no more coda errors*

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

There is no sense in which a BCD learner could choose to install Max above only NoCoda and not *ComplexCoda on the basis of the error in (2a). However, this is

precisely the kind of intermediate stage that learners of languages with complex codas (like English) do go through, often for several months of development (see §1.1.1).

The illustration in (2) shows that to avoid learning complex codas immediately from just one word like 'toast', our learner must either not learn using BCD, or it must not learn from errors like (2a). Since the previous chapter was dedicated to the claim that BCD is a successful way to ensure that learners reach the right end state grammar, the proposal I will make in this chapter takes the latter approach and provides a more gradual way in which errors enter the Support and so trigger re-ranking.

Briefly, the idea is this. In Error-Selective Learning, errors are not immediately added to the Support when they're made, but rather stored temporarily until a sufficient number of ERC rows have demonstrated one particular problem with the current ranking. Only at that point does the learner choose to update the Support – and not with all errors made, but with just an error that will cause a minimal change to the grammar.

The criteria that choose the right error to add to the Support are designed with two particular kinds of intermediate stage in mind. In the following two sections (§1.2 and §1.3), I provide a representative example of each from the literature; many more examples of each appear in section 2 below for the data-interested reader. I acknowledge in advance that the two types of intermediate stages I discuss below do not exhaustively describe every such attested stage in the acquisition literature – and in fact, that there are patterns found commonly in development that will *not* be captured by the Error-Selective idea.[1] I focus here on these two stages in an attempt to make some initial progress, but the

[1] One aspect of phonological development I will have nothing to say about here is the role of child-specific templates in e.g. Priestly (1977), Vihman and Croft (**XX**), Vihman (**XX**) and references therein. These are templates that a particular child adopts at an intermediate stage which overwrites segmental material predictably across a range of lexical items, but which unlike most templatic effects in adult and other child

ultimate success of the Error-Selective model will have to be judged with respect to a much broader range of data.

1.2 The Specific Markedness stage: English coda clusters

The data below from Trevor (Compton and Streeter 1977 – see §2.1.1) demonstrate the claim made above that English-learning children often pass through three stages of coda acquisition. At the first stage no codas are produced, at the intermediate stage only singleton codas are produced,[2] and at the final stage complex codas now also appear.

3) *Trevor's three stages of coda acquisition*

a) All codas deleted (up to 1;4.2)

singleton codas			*complex codas*		
Target	*Child*	*Age*	*Target*	*Child*	*Age*
'duck'	[dʌ]	0;10.17	'plant'	[tɛ]	1;3.11
'cup'	[kʌ]	1;1.0	'orange'	[oŋ]	1;4.2
'puppet'	[pʌpə]	1;3.25			

b) **Intermediate stage**: singleton codas only (1;5-1;7.26)

singleton codas			*complex codas*		
Target	*Child*	*Age*	*Target*	*Child*	*Age*
'walk'	[wɔk]	1;6.8	'box'	[gak]	1;7.11
'hat'	[hæt]	1;6.8	'toast'	[toːs]	1;7.20
'melon'	[mɛːmɪn]	1;7.26	'milk'	[mʊːk][3]	1;7.26

phonology do not seem to be motivated by markedness pressures: c.f. McCarthy and Prince (1993); Gnanadesikan (2004). There is also the tricky issue of apparent phonological regressions, which are at least partially addressed in §6.

[2] This is a simplification of Trevor's singleton coda development. I am abstracting away here from the fact that he actually appears to learn stressed codas before unstressed ones. See section 2.3.1 for more.

[3] This lengthening of the vowel does not appear to be a consistent mapping for dark [l]; many other words at this stage are transcribed with vowel lengthening that does not correspond to any missing input segment, and other missing [l]s do not trigger vowel lengthening.

c) All codas retained (1;9 onwards)

singleton codas			*complex codas*		
Target	*Child*	*Age*	*Target*	*Child*	*Age*
'room'	[wuːm]	1;9.2	'plant'	[pænt]	1;9.2
'egg'	[ɛg]	1;9.28	'stairs'	[ʃtɛrʃ]	1;9.2
'outside'	[saiːd]	1;9.28	'toast'	[toːst]	1;9.29

The intermediate stage in 3b) is one which requires a ranking like in 4) below:

4) *A Specific Markedness stage*
*ComplexCoda >> Max >> NoCoda

As the simple tableaux below illustrate, this ranking protects singleton codas, but still reduces complex coda clusters:

5)a) Max >> NoCoda protects singleton codas in 'walk'

/wɔk/	Max	NoCoda
☞ [wɔk]		*
[wɔ]	*!	

5)b) NoComplex >> Max reduces coda clusters in 'toast'

/tost/	NoComplexCoda	Max	NoCoda
[tost]	*!		*
☞ [tos]		*	*
[to]		*!*	

1.3 The Specific Faithfulness stage: French onset clusters

The second kind of intermediate stage comes from Rose (2000), who documented stages in the acquisition of Québécois French by two children, Clara and Théo. He presents evidence of a stage at which complex onsets are preserved faithfully in stressed syllables, but the same clusters are reduced to singleton in unstressed syllables:

6) *Clara's three stages of onset acquisition* (see Rose 2000:130-133)

a) All onsets reduced (1;0.28-1;09.01)

stressed syllable			*unstressed syllable*		
Target	*Child*	*Gloss*	*Target*	*Child*	*Gloss*
/kχa.ˈkχa/	[ka.ˈkæ]	'Cracra' (name)	/bʁiˈze/	[bœˈçiː]	'broken'
/plœʁ/	[pœː]	'(s/he) cries'	/abʁiˈko/	[pupæˈko]	'apricot'
/flœʁ/	[βœː]	'flower'			

b) Intermediate stage: stressed onsets retained (1;09.29-2;03.05)

stressed syllable			*unstressed syllable*		
Target	*Child*	*Gloss*	*Target*	*Child*	*Gloss*
/bi.ˈbʁɔ̃/	[pa.ˈpχɔ]	'baby bottle'	/fʁi.ˈgo/	[bʊ.ˈko]	'fridge'
/glɪs/	[klɪs]	(he/she)'slides'	/bʁy.ˈle/	[bɪ.ˈle]	'burned'
/si.ˈtχʊj/	[θə.ˈtχʊːj]	'pumpkin'	/gli.ˈsad/	[ka.ˈsæd]	'slide'
/plœʁ/	[plœχ]	'(he/she)cries'	/tχu.ˈve/	[tʊ.ˈve]	'found'

c) All onsets retained (2;03.15 onwards)

stressed syllable			*unstressed syllable*		
Target	*Child*	*Gloss*	*Target*	*Child*	*Gloss*
/ˈgʁo/	[ˈgʁo]	'big'	/tχu.ˈve/	[tχʊ.ˈve]	'found'
			/plɑ̃.ˈʃe/	[plɑ.ˈʃe]	'floor'

The ranking this 6b) intermediate stage suggests:

7) *A Specific Faithfulness stage*
Max-(σ') >> *ComplexOnset >> Max

In this ranking, markedness is sandwiched between two faithfulness constraints – the higher-ranked of which is a positional version of Max. I will return to the correct definition of this constraint in section 1.4, but its effect will be to prevent input segments from being deleted from stressed syllables. With this constraint, we get Clara's intermediate stage, as below:

8)a Max σ' >> *ComplexOnset protects clusters in stressed syllables

/glɪs/	Max-σ'	*ComplexOnset
klɪs		*
kɪs	*!	

8)b *ComplexOnset >> Max reduces clusters elsewhere

/gli.ˈsad/	Max-σ'	*ComplexOnset	Max
kla.ˈsæd		*!	
ka.ˈsæd			*

It should be noted that the specific faithfulness constraints that I focus on in this chapter are positional rather than featural – that is, the contexts in which they apply are prosodic categories higher than the segment (stressed syllables, initial syllables, and Roots). However, many OT analyses of intermediate stages in child development use Ident[feature] constraints in ways that, if translated into a *stringent* theory of featural faithfulness, would provide parallel rankings to the one in (7). (For examples, see Pater (1997) section 4.3 on child consonant harmony constraints intermingled with faithfulness and the grammar fragments in Pater and Barlow (2003) figures 2-4.)

1.4 Analytic assumptions about the intermediate stages

1.4.1 Stringency relations among markedness constraints

Chapter 2 already made it clear that the theory of faithfulness I adopt uses stringency relations to capture the positional and specific contexts of faithfulness constraints. In the coda vs. complex coda example from 1.2 above, I've similarly characterized the "flanking" markedness constraints as being in a stringency relation. The top Markedness constraint is *ComplexOnset, given as a separate constraint from *ComplexCoda – rather than using a single *Complex constraint which would not be in a

stringency relation with NoCoda (because CCVC syllables would violate the former and not the latter.).

In this particular case the split seems well-justified, if only because children acquire clusters edge by edge (see the data above, or the Dutch data presented in Levelt and van der Vijver, 2004). There might well be different (or even better) analyses of the data that I present in this chapter that does not involve stringency relations between its constraints – and certainly there are intermediate stages that do not include such stringency relations. In the end, stringency relations between markedness constraints will turn out to be a relevant but not necessary aspect of my Error-Selective method of deriving stages. However, they do provide a clear example of how the proposal works.

1.4.2 Positional faithfulness and input prosodic structure

Two kinds of positional faith constraints are used in this dissertation to capture intermediate stages of acquisition. The first is the more common positional Ident constraint used normally in analyses of adult grammars (see the body of references in chapter 1 §4.1). As we already saw in chapter 1, these constraints retain input properties in a particular output context, e.g.:

9) Ident[voice]-Onset: "Output segments syllabified as onsets must match their input correspondents for the feature [voice]"

The second kind of positional faithfulness constraint, which in fact play a role in the majority of this section's analyses, are Max constraints with a positional flavour. These constraints prohibit deletion only in certain output contexts. To make coherent the

notion of "deleting from an output context", these constraints must be defined across inputs and outputs which *both* contain prosodic contexts.[4] Thus the Max-σ' constraint used in section 1.3 will be defined as:

10) Max[Seg]-Onset: "Input segments syllabified as onsets must have output correspondents"

These constraints are therefore only violated when segments are already syllabified as onsets in the input and then deleted in the output.[5]

As a consequence of their definition, these positional Max constraints require the assumption that the learning child has prosodic structure in his or her *inputs* as well. In the French case of complex onsets above: Clara must first have gotten the French syllabification right, to have parsed word-medial obstruent-liquid clusters as complex onsets. But she must also have encoded that syllabification in the input itself, so that the constraint in (10) protects her input stressed syllables.

The problem with assuming input prosodification and positional Max constraints of this sort is that they will predict an unattested set of languages with contrastive syllabification – e.g. a language that contrasts [pa.ta] and [pat.a] to remain faithful to input syllable structure (for discussion of this point, see e.g. McCarthy 2006). This state of affairs presents a dilemna: on the one hand it appears that adult grammars do not

[4] In fact, positional Ident constraints may need to refer to input prosodification in order to account for a wider range of phenomena than those discussed in this chapter -- see especially Wilson (2000) – although see McCarthy (2006) for a solution to these problems for Ident that involves a different notion of GEN (OT with Candidate Chains.) Whether an OT-CC approach can also provide a solution to the problem with positional Max raised in the main text above is an interesting question for further consideration.

[5] Note that Beckman (1998) chapter 5 defines Positional Max rather differently; see also Alber (2001).

syllabify contrastively, while on the other hand child grammars DO require positional constraints that ban deletion (see the data §2.3 below).

To resolve this conflict, something learning-specific must be said about positional Max constraints.[6] One possible approach would be to say that learners are prevented from positional deletion by *output-output* faithfulness relations of some sort – e.g. requiring segments in strong contexts of the *winners* to be retained in the *losers*.[7] In this way, we might re-define the Max[Seg]-Onset constraint in 10) above as 11):

11) OO- Max[Seg]-Ons: "*Winner* segments syllabified as onsets must have output correspondents."

Such positional Max constraints would avoid any typological problems among adult grammars, because they can only assess violations in a tableau when one candidate is designated the *winner*.[8]

To prevent distraction from the central points of the chapter, I will not adopt this OO-faith definition in 11) in the tableaus to come, and instead will add prosodic structure to the learner's inputs when necessary. However determining the correct definition of positional Max constraints and their precise learning-specific properties are crucial issues for future research, especially because they raise important questions about the precise status of winners and losers and the relationships between developing and end-state grammars.

[6] Cf. the discussion of this approach by Rose (2000); see also the child-specific syllabic structures assumed by Goad and Rose (2004).
[7] Thanks to John McCarthy for initial discussion of OO-faith's role in defining these constraints, and to Della Chambless and Joe Pater for later discussion.
[8] Assessing this constraint is admittedly not a trivial matter – it would mean understanding 'winner' as something like a Fully-Faithful Candidate (see McCarthy 2006) that stands in correspondence with every other candidate. This is also reminiscent of the sympathetic candidate, in Sympathy Theory approach to opacity (McCarthy 1999)

1.5 Roadmap to the chapter

To provide some fodder for the proposal, section 2 below provides a small range of examples of the two kinds of intermediate stage discussed above, from both the existing literature and some of my own corpus work. Section 2.2 concentrates on a couple of Specific-M cases; section 2.3 spends some more time introducing a range of Specific-F effects. (The less data-minded reader can skip this section without theoretical repercussions.) Section 3 then introduces the Error-Selective Learning technique: spelling out the procedure in §3.1, discussing some of why and how it works in §3.2, and exemplifying its use in a case study of two children's complex onset acquisition in §3.3.

Section 4 turns to a more thorough discussion of one aspect of Error-Selective Learning, namely the way its stages are connected to input frequencies. I discuss the frequency predictions of ESL (§4.1), and some evidence for the predictions from the literature (§4.2), including predictions that are not strictly tied to the Specific-M and Specific-F stages (§4.3). I also point out that ESL's partial reliance on error frequency makes the BCD learner robust to noisy data. Section 5 makes some speculative proposals about how the Error-Selective CD learner could be extended to model variation between stages of acquisition, and section 6 concludes.

2. The data from intermediate stages

2.1 Introduction to the data

The data discussed in this chapter has been selected both to demonstrate the kinds of intermediate stages that Error-Selective learning can handle, but also to provide samples of representative intermediate stages in the literature to date. Much of the chapter focuses on the acquisition of syllable structure: codas and coda clusters, onset clusters

and other syllable margins, although examples are also drawn from the development of word shape (i.e. stages of syllable truncation). Later, in chapter 4, I will also turn briefly to some attested stages of morpho-phonological development (chapter 4 §7.2; see also this chapter §2.3.4).

I note here that the stage-by-stage characterization of the data in this section will abstract away from any and all variation within stages (although clearly the quantitative results I provide and cite from others will belie this idealization.) I leave the discussion of variation to section 5.

2.1.1 The Compton/Streeter database

Much of the data discussed in this chapter is taken from existing sources (see references with each example) but I also use data from my own work on the corpus of two children, Trevor and Julia. These data are taken from the Compton/Streeter/Pater database (Compton and Streeter 1977; Pater 1997; Pater and Werle 2001; Pater and Barlow 2003.) This database contains transcriptions by the children's mothers, who were speech pathologists and had received additional training in child transcription before beginning the data collection. Data on the amount and breadth of the data from the children is provided below in 12):

12) *Statistics about Trevor and Julia's corpora*

	first and last session	total no. of tokens[9]
Trevor	0;8 – 3;1.8	12,177
Julia	1;2 – 3;1.3	5,772

[9] Note that some 'tokens' are really utterances, so that one 'token' might include more than one word in the data reported below.

Table 13) provides the number of different words that each child had used by the end of each month they were followed:

13) *Total lexical items*

up to…	*Trevor*	*Julia*
1;3	78	--
1;4	129	30
1;5	196	52
1;6	297	112
1;7	374	164
1;8	440	232
1;9	484	309
1;10		379

The data for these two children seems rather comprehensive – that is, it would appear to contain nearly all the words (types) Trevor and Julia uttered, up until near the end of their second year (1;9 or 1;10).

2.2 Intermediate stages that rely on specific markedness

2.2.1 More on complex codas in Germanic

It is well-established that children acquiring languages with complex coda structures (like English and Dutch) often go through an intermediate stage where singleton codas are preserved faithfully, while coda clusters are reduced to singletons.

14)

The initial stage:		*The intermediate stage:*	
/CVC/ →	[CV]	/CV<u>C</u>/ →	[CV<u>C</u>], *[CV]
/CV<u>CC</u>/ →		/CV<u>CC</u>/ →	[CV<u>C</u>], *[CVCC]

This stage was exemplified with some of Trevor's data in section 1; in (15) below I provide additional examples from other children at this intermediate stage. The Dutch

data from Eva come from Fikkert (1994), Levelt (1994); G's English data come from Gnanadesikan (1995/2004), and P.J.'s English data are from Demuth and Fee (1995):

15) *The intermediate stage of coda acquisition*

	singleton codas retained			*complex codas reduced*		
	Target	*Child*	*Gloss*	*Target*	*Child*	*Gloss*
Dutch: Eva (1;4,12)	/teːn/	[teːn]	'toe'	/eːnt/	[eɪn]	'duck'
	/bɛd/	[dɛt]	'bed'	/staːrt/	[tat]	'tail'
English: G (2;3-2;9)	/greɪp/	[geɪp]	'grape'	/drɪŋk/	[bɪk]	'drink'
	/piz/	[piz]	'peas'	/frɛnd/	[fɛn]	'friend'
	singleton codas retained (or deleted)			*complex codas reduced (or deleted)*		
English: PJ (1:11)	/wak/	[rɑʔ]	'walk'	/tost/	[toːs]	'toast'
	/sup/	[sup]	'soup'	/bidz/	[biːs]	'beads'
		[suː], [su]			[beː]	
	/dʒus/	[dʒuːs], [duːs]	'juice' [10]			
		[dʒuː]				

Again: this pattern is derived by the ranking in 16):

16) *ComplexCoda >> Max >> NoCoda

2.2.2 Markedness of complex onsets, and sonority distance

A different example of a stage that requires this kind of ranking comes from the development of onset clusters. Along the developmental path from singleton onsets to the full English set of complex onsets, Trevor and Julia both go through stages where stop-r

[10] In Demuth (1996) the child pronunciation AND adult target 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.

and stop-glide onsets consistently surface faithfully, but stop-l onsets are almost always reduced to singletons:

17) *The initial stage:* all /CVC/ → [CVC]

The intermediate stage: /CrVC/, /CwVC/ → [CCVC], *[CVC]; /ClVC/ → [CVC], *[ClVC]

As summarized in 18), Trevor's initial stage lasts until 2;2:

18) Trevor's initial stage: all onset clusters reduced (up to 2;2)

age	*complex onset inputs*			
	outputs raw #s		*output percentages*	
	[C]	[CC]	[C]	[CC]
up to 1;7	344	3	**99.1**	0.9
1;8-2;2	838	130	**86.6**	13.4

19) Trevor's representative words at stage one

stop-liquid clusters: reduced			*stop-r (and stop-glide)* [11] *clusters: retained*		
Target	*Child*	*Age*	*Target*	*Child*	*Age*
'blocks'	[gak]	1;5.18	'cracker'	[kaka]	1;5.14
'clock'	[kak]	1;6.17	'train'	[teːn]	1;5.14
'glasses'	[ˈgæʃɪʃ]	1;8.12	'brush'	[baɪʃ]	1;6.25

After leaving the initial stage, Trevor then went through two months (2;3-2;4) at the intermediate stage discussed above. I have broken down Trevor's cluster treatment at this point into three categories: stop-r, stop-l and tr. I leave out tr clusters below, because Trevor's /tr/ cluster has something of an independent trajectory that seems mostly due to the [tʃr]-initial pronunciation of his own name that he adopts at this point.

[11] I have not included raw numbers from Trevor's input stop-w words at this stage because they are so few.

20) Trevor's intermediate stage: stop-r retained more than stop-l (2;3-2;4)[12]

		raw #s		*percentages*	
age	*output*	*stop-l*	*stop-r*	*stop-l*	*stop-r*
2;2	C…	30	34	**100.0**	**82.9**
	CC…		7	0.0	17.1
2;3	C…	23	12	**69.7**	31.6
	CC…	10	26	30.3	**68.4**
2;4	C…	28	18	**62.2**	41.9
	CC…	17	25	37.8	**58.1**

21) Trevor's representative words at the intermediate stage:

stop-liquid clusters: reduced			*stop-r/stop-glide clusters: retained*		
Target	*Child*	*Age*	*Target*	*Child*	*Age*
'glass'	[ga:s]	2;3.22	'grapes'	[grepts]	2;3.4
'play'	[pei]	2;3.30	'quite'	[kwait]	2;3.4
'cleaner'	[ki:nə]	2;4.13	'present'	[pwɛsɛn]	2;4.3

Julia (who talks much less than Trevor, but also begins to produce complex onsets much earlier) is at the initial stage of onset acquisition up until about the end of 1;9:

22) Julia's initial stage: up until 1;9.25

age	*complex onset inputs*			
	output raw #s		output percentages	
	[C]	[CC]	[C]	[CC]
up to 1;7	55	0	**100.0**	0.0
1;8-1;9	110	28	**79.7**	20.3

23) Representative words at Julia's stage one

stop-liquid clusters			*stop-r and stop-w clusters*		
Target	*Child*	*Age*	*Target*	*Child*	*Age*
'please'	[pɪs]	1;7.7	'cry'	[kai]	1;7.9
'blankie'	[bæki]	1;6.15	'drive'	[waiv]	1;9.14
'clown'	[kaʊn]	1;8.25	'truck'	[fʌk]	1;9.25

[12] Two notes about this stage. First, it is sometimes Dep that gets violated in the case of stop-liquid clusters, rather than Max: 'problem' as [pwa:bəlem], (2;3.7). Second, Trevor goes through a brief period at 2;3 where /pl/ in particular is preserved as [pw], but then he reverts again to singleton [p].

Julia's intermediate stage lasts from about 1;10-2;1. At stage two I have broken down her clusters into stop-r, stop-l and br. This last cluster is separate because [br]'s acquisition is delayed compared to all other stop-r clusters (see the final two columns table of 24) below.)[13] I have also included the numbers from 2;2, which show she's on her way to a stage where all these clusters surface:

24) Julia's intermediate stage: stop-r clusters retained while stop-l reduced (1;10-2;1)

		raw #s		*percentages*			
age	*output*	*stop-l*	*stop-r/w*[14]	*stop-l*	*stop-r*	*br*	*%*
1;10	C…	21	9	**95.5**	26.5	4	
	CC…	1	25	4.5	**73.5**	0	
1;11	C…	22	1	**100.0**	2.4	4	
	CC…	0	40	0.0	**97.6**	2	
2;0	C…	23	3	**100.0**	7.9	5	
	CC…	0	35	0.0	**92.1**	0	
2;1	C…	17	2	**89.5**	5.4	4	
	CC…	2	35	10.5	**94.6**	0	
2;2 (beginning of next stage)	C…	0	5	0.0	10.4	0	
	CVCV	6	0	21.4	0.0	6	
	CC…	22	43	**78.6**	0	**89.6**	

25) Representative words from Julia's intermediate stage

stop-liquid clusters: reduced			*stop-r/-glide clusters: retained*		
Target	*Child*	*Age*	*Target*	*Child*	*Age*
'glasses'	[gaθəs]	1;10.10	'drink'	[gwɪŋk]	1;10.5
'please'	[pɪs]	1;10.10	'queen'	[gwɪn]	2;0.2
'clap'	[kæp]	1;11.15	'crown'	[kwaUn]	2;0.2

[13] With respect to /br/: given the fact that she pronounces many, if not most, of these dependent onset /r/s as [w]s, one reasonable explanation is that her grammar rules out an onset cluster parse /br/ → [bw] via the OCP-labial constraint that independently rules out [bw] and [pw] in English.

[14] During this stage, Julia uses what is transcribed as [fw] for /sw/ and occasionally other clusters like /kw/. Noting that all these input clusters are voiceless, and include both labial and velar place, Joe Pater (p.c.) suggests Julia is actually producing a voiceless labio-velar consonant, having fused these clusters' features into a single segment.

This stage can be derived by ranking faithfulness (Max-Seg) between some two markedness constraints that affect complex onsets. Here I will suggest that the relevant constraints are on the relative sonority of the onset's segments. Below I spend some time on the theoretical details of these constraints because they will be used in the case study in section 3.3.

The constraints I will use here are an adaptation of Baertsch (2002)'s Split Margin Hierarchy constraints, which penalize onset clusters (among others) according to the sonority of first and second members.[15] The sonority hierarchy I adopt is in 25) (see e.g. Blevins, 1995; Clements, 1990; Murray and Venneman, 1983; Parker, 2002; also Pater and Barlow, 2003):

26) *Relevant Sonority Hierarchy (from most to least sonorous)*
vowels > glides > r > l > nasals > fricatives > stops

Note that I have adopted a sonority hierarchy that distinguishes [r] as more sonorous than [l], precisely because it is this difference that matters here.[16]

Two well-known generalizations are (a) onset clusters rise in sonority, so that their first member is less sonorous than their second, and (b) the larger the rise in sonority, the better the cluster.[17] To capture these generalizations, Baertsch (2002) expands on Prince and Smolensky (1993)'s Peak and Margin hierarchies and uses them similarly to build constraints which each penalize a sequence of sonority levels. Baertsch

[15] See also Gouskova (2001) for a similar hierarchical OT approach to the markedness of cluster sonority, but for coda-onset sequences.
[16] One argument for r's higher sonority than l comes from English rime structure, on the assumption that the more sonorous a segment the better a nucleus it makes. While some English speakers claim to have monosyllables in 'earl' and 'squirrel', none report the intuition of a monosyllabic [lr] rime sequence. See also Parker (2002).
[17] at least to a point; see e.g. Clements (1990), in which the best sonority distance is calculated both between onset segments and between those segments and the following vowel nucleus.

then organizes her constraints into fixed rankings to reflect the second generalization above – the less sonority rises between the first and second members of an onset cluster, the more marked the cluster is, and the higher its constraints sit in the fixed ranking. This is illustrated in 27) below – I have adopted Gouskova (2004)'s adaptation of the constraints, though using the reduced sonority scale of 26) above. In these constraints, 'T' stands for any stop, 'S' for any fricative, 'N" for any nasal, 'L' and 'R' for themselves, and 'W' for any glide:

27) *Onset Sonority Distance Hierarchy (Baertsch, 1998, 2002; from Gouskova, 2004)*
*WT>>{*WS,*RT}>>…{*WW,*RR,*LL,*NN,* SS,*TT}>>…{*SW,*TR}>> *TW

Thus, the most marked onset cluster is one which maximally *falls* in sonority – $_{\sigma}$[wta], meaning in this case any glide followed by any stop – and the least marked is one which maximallly *rises* – $_{\sigma}$[twa].

From the present perspective, I translate the constraints in (27) in the fixed hierarchy into a set of *stringent* constraints. These stringent constraints each ban a particular point on the onset sonority distance hierarchy, as well as anything above (i.e. more marked than) that point. This is shown in the prose definitions of these constraints below; note that not every point on the scale is illustrated on this scale (skipped constraints are indicated by the ellipses):

28) *Stringent Onset Sonority Distance Constraints (built from 27)*

(a) *WT = "No glide-stop onsets"

(b) *WS, RT = "No glide-fricative or r-stop (*or glide-stop*) onsets"

(…)

(c) *WW,RR,LL,NN, SS,TT
= "No onsets with a sonority plateau
(*or with any sonority drop*)"
(…)

(d) *LW, NR, SL, TN = "No liquid-glide, nasal-liquid, fricative-liquid or stop-nasal onsets (*or anything with less or a sonority rise)*"

(e) *NW, SR, TL = "No stop-liquid, fricative-r, or nasal-glide onsets (*or anything with less of a sonority rise)"*

(f) *SW, TR, = "No fricative-glide or stop-r onsets
(*or anything with less of a sonority rise*)"

(g) *TW = "No stop-glide onsets
(*or anything with less of a sonority rise*)"
i.e.,"No complex onsets"

For the case we are analyzing here, the relevant two constraints are (28e) and (f); these are near the most stringent end of the constraint set, meaning they penalize all but the best clusters. Since the data we are focusing on here is just *stop-initial* clusters, we can reword the effects of these two constraints as in (29):

29) *Stop-initial clusters allowed by two onset sonority constraints*

a) *NW, SR, TL obeyed by stop-r and stop-glide clusters
(abbreviated to ***TL)**

b) *SW, TR, obeyed by stop-glide clusters
(abbreviated to ***TR**)

With Max-Seg between these constraints, we get the right effect:

30) *TL >> Max[Seg] >> *TR

To illustrate using Julia's data from 25):

31) Max >> *TR protects the onset cluster in 'drink'

/[drɪŋk]/	Max	*TL
☞ [gwɪŋk]		*
[gɪŋk]	*!	

32) *TL >> Max reduces the cluster in 'please'

/[plɪs]/	*TL	Max	*TR
[plɪs]	*!		*
☞ [pɪs]		*	

In this theory, the constraint *ComplexOnset can be dispensed with – or rather, it is simply equivalent to the most stringent onset sonority constraint, *TW. However, in all subsequent tableaux when onset sonority is not at issue, I will continue to use the simple constraint label *ComplexOnset.

2.3 Intermediates stages that rely on specific faithfulness

This section discusses developmental stages that are best (or only) analyzed with reference to positional faithfulness constraints – and in particular, positional Max.

2.3.1 More on faithfulness in stressed syllables

The first case, already discussed in section 1, comes from Rose (2000). He presents evidence of a stage at which complex onsets are preserved faithfully in stressed syllables, but the same clusters are reduced to singleton in unstressed syllables (see Rose 2000:130-133):

33) *the initial stage*
/…CCVC…/ → […CVC…]

the intermediate stage
/CV.ˈ**CCV**/ → [CV.ˈ**CCV**], *[CV.ˈCV]
/**CCV**.ˈCV / → [**CV**.ˈCV], *[CCV.ˈCV]

34) *Théo's initial stage of onset cluster acquisition*[18]

a) Stage one: all onset clusters simplified (up to 2;05.11)

stressed syllables			*unstressed syllables*		
Target	Child	Gloss	Target	Child	Gloss
/klʊn/	[kʊɲ]	'clown'	/bʁiˈze/	[piˈze:]	'broken'
/tχɛ̃/	[kɛ]	'train'			

35) **Intermediate stage**: complex onsets in stressed syllables only (2;05.29 - 2;11.29)

stressed syllables: retained			*unstressed syllables: reduced*		
/gʁo/	[gʁo]	'big'	/tχak.tœʁ/	[taˈtœᵘ]	tractor
/tχɛ̃/	[kχɛ]	'train'	/gʁy.jo/	[kʰœ.ˈjɔ]	oatmeal
/kle/	[kxi]	'key'	/tχu.ve/	[kʊ.ˈβi]	found
/plœʁ/	[plœᵘ]	'(he/she)cries'	/ˌkχɛm.gla.ˈse/	[ˌkχa'..na.ˈse]	ice cream

36) Stage three: all onset clusters retained (3;0.7 onwards)

stressed syllable			*unstressed syllable*		
Target	*Child*	*Gloss*	*Target*	*Child*	*Gloss*
(none cited)			/tχu.ˈve/	[kχa.ˈve]	'found'
			/pχə.ˈne/	[pʁə.ˈne]	'(you.pl) take'
			/plœ.ˈʁe/	[plø.ˈʁe]	'to cry'
			/gli.ˈsad/	[kli.ˈsad]	'slide'

As already discussed, the ranking that accounts for this intermediate stage is:

37) Max-stressed-σ >> *Complex >> Max-σ

(For evidence of this ranking in adult language, see Goad and Rose (2004) on Brazilian Portugese.)

Similar data comes from Kehoe and Debove-Hilaire (2003), but from the acquisition of C-glide rather than C-liquid sequences. The 14 children in their experiment (ages 1;10 -2;9, mean age 2;4) preserved the CGV structure of CwV and CHV more often in stressed than unstressed syllables ($p < 0.01$). For two children, the effect was nearly-

[18] See Rose (2000) section 3.4.1 for the full details of Théo's stop-liquid acquisition

categorical, in that CGV sequences were retained 100% of the time in stressed syllables, but less than 20% of the time in unstressed syllables.

Lléo (2003) and Pietro and Bosch-Baliarda (200**X**) provide similar support for activity of stressed-syllable faith with respect to the development of codas, in Spanish and Catalan respectively. When Lléo's subjects, José, first begins to produce codas between 2;0-2;2, he does so only in stressed syllables:

38) *the initial stage*
/CVC.CVC/ → [CV.CV]

the intermediate stage
/C**V**C.CV**C**/ → [C**V**C.CV]
/CV**C**.C**V**C/ → [CV.C**V**C]

The two tables below that chart these two stages are adapted from Lléo (2003):

39) José's stage one: no consonantal codas produced (1;7-1;11)

a) final codas

	stressaed σ			unstressed σ		
age	targets	faithful	% faith	targets	faithful	% faith
1;7	22	1	5%	1	0	0%
1;9	*29*	*7*	*24%*	6	0	0%
1;10	10	0	0%	5	0	0%
1;11	21	0	0%	10	0	0%
total	145	18	12.4%	48	1	2.1%

b) medial codas

	stressaed σ			unstressed σ		
age	targets	faithful	% faith	targets	faithful	% faith
1;7	8	3		3	0	0%
1;9	34	1	3%	16	1	6%
1;10	63	12	19%	16	0	0%
1;11	77	4	5%	25	3	12%
total	344	69	20.1%	132	11	8.3%

Note that at 1;9, all the "codas" that José is producing are *vowels*. It should be noted that Lléo does report explicitly that these vowels only ever appear as coda substitutes, and that he is not merely at a stage where vowel length/quality is uncontrolled. Nevertheless: while it may be that these epenthetic vowels are related to the input codas, they still do not indicate mastery of a stage where NoCoda has been demoted. Instead, this begins at 2;0 and increases considerably at 2;2:

40) Jose's stage two – 2;0-22[19]

a) final codas

	stressaed σ			unstressed σ		
age	targets	faithful	% faith	targets	faithful	% faith
2;0	31	4	13%	14	1	7%
2;2	**30**	**6**	**20%**	11	0	0%
total	145	18	12.4%	48	1	2.1%

b) medial codas

	stressaed σ			unstressed σ		
age	83	12	14%	39	4	10%
2;0	**74**	**35**	**47%**	32	3	9%
2;2	344	69	20.1%	132	11	8.3%
total	83	12	14%	39	4	10%

Thus we have the ranking in 41); the tableaux in 42) below use data from Lléo (2003) table 2 to demonstrate the ranking:

41) Max[Seg]-stressed-σ >> NoCoda >> Max[Seg]

[19] As Lléo (2003) rightly notes, José's production of codas in medial position is better than in final position – something I have nothing say about here.

42) NoCoda >> Max reduces the coda in *dos* 'two' (1:7.27)

/[dos̲]/	NoCoda	Max
[dos]		*
☞[dœ:]	*!	

43) But Max-σ' >> NoCoda retains the coda in *venga,* 'come on' (1;10.3)

/[bɛn̲ga]/	Max-σ'	NoCoda	Max
☞ [bɛn̲ga]		*	
[bɛga]	*!		*

This stage also appears in Trevor's data. During his first months of solid coda production, his outputs almost always retain codas only in stressed syllables. Given the difference in frequent word shapes of English vs. Spanish, I characterize his intermediate stage slightly differently than José's, but the ranking remains the same:

44) *the initial stage*
/CV**C̲**/ → [CV]
/CVCV**C̲**/ [CVCV]

the intermediate stage:
/C**V**C̲/ → [CVC̲], *[CV]
/CVC**V**C̲/ → [CVC̲], *[CV][20]
/C**V**CVC̲/ → [CVCV], *[CVCVC̲]

45) Trevor's stage one: no codas anywhere (up to 1;3)

stressed syllables			*unstressed syllables*		
Target	*Child*	*Age*	*Target*	*Child*	*Age*
'duck'	[dʌ]	0;10.17	'puppet'	[pʌpə]	1;3.25
'cup'	[kʌ]	1;1.0	'orange'	[oŋ]	1;4.2

46) Trevor's intermediate stage: singleton codas only in stressed syllables (1;5-1;6)

stressed syllables			*unstressed syllables*		
Target	*Child*	*Age*	*Target*	*Child*	*Age*
'all gone'	[gɔˈgɔn̲]	1;5.4	'puppet'	[pʌpə]	1;5.5
'bike'	[gak̲]	1;5.30	'blanket'	[kækɪ]	1;5.14
'hat'	[hæt̲]	1;6.8	'yogurt'	[gogə]	1;5.30

[20] Truncation due to e.g. Trochee and Parse-σ

2.3.2 Faithfulness *to* stressed syllables

There are also several examples from the literature on syllable truncation where children resist the pressure to delete syllables from a privileged (stressed or initial) position.[21]

With respect to the stressed syllable position, Kehoe and Stoel-Gammon (1997) and Kehoe (2000) report on an elicitation study of English-speaking children at (2;4) and (2;10), designed in part to test for stress effects on syllable truncation. In their data, truncation patterns were almost exclusively restricted to unstressed syllables while stressed ones were retained. The most compelling evidence for Max-σ' in this data comes from a stage in 47) below, from Kehoe (2000)'s section 4.2.2:

47) *the English intermediate stage*
/wSw/ → [(Sw)] /SwSw/ → [(S)(SW)]

48) *the data* (taken from Kehoe (2000) tables 4,7, 10)[22]

	Subject	*Target* **/wSw/**	*Child* **[Sw]**	*Target* **/SwSw/**	*Child* **[SSw]**
a)	22m3	banana	[nǽnʌ], [nǽŋʌ]	àlligátor	[ǽgè.ʌ]
b)	22f1	banána	[nǽnʌ], [nǽnə]	àlligátor	[ǽgædə˞]
c)	27m6	banána	[báni]	àvocádo	[ɑ̀kádo]
d)	28f2	banána	[bǽ:nʌ]	àvocádo	[ʌ̀kádo]

As Kehoe points out, output forms like [(ɑ̀)(kádo)] for "avocado" include a marked initial degenerate foot (ɑ̀). The explanation for why these children's grammars preserve the first syllable of "avocado" but still truncate the first syllable of "banana" must therefore be a pressure to retain stressed syllables only, at the expense of marked foot shape. Thus, this stage provides evidence of another Specific-F ranking, namely:

[21] Recall from section 1.4.2 that this analysis assumes the positions exist in the *input* as well.
[22] Since this was a cross-sectional study, we don't have the data to show any of the earlier truncation stages these children were at: e.g., one where outputs were always one foot (see §4.3.)

49) Max[Seg]-σ' >> *DegenerateFoot[23] >> Max[Seg][24]

50) Markedness rules out a one-syllable initial foot in 'banana'

/bənǽnʌ/	*DegenerateFoot	Max[Seg]
☞ (nǽnʌ)		*
(bə)(nǽnʌ)	*!	

51) Max(Seg)-σ' protects the initial syllable of 'avocado'

/ɑ̀vəkádo/	Max[Seg]-σ'	*DegenerateFoot	Max[Seg]
(kádo)			*
☞(ɑ̀)(kádo)		*!	

2.3.3 Faithfulness to initial syllables

One piece of evidence of an intermediate stage that relies on *initial* syllable faith comes from a later stage of syllable truncation in Greek (Revithiadou and Tzakosta, 2004); there is also some evidence for this stage in Spanish mentioned in Gennari and Demuth (1997). The former authors provide evidence from four different children acquiring Greek at a stage where words with more than 3 syllables get reduced to the stressed-foot, *plus the input's initial syllable* – while other unstressed syllables are reduced:

[23] This choice of markedness constraint is not the only option; each option will probably require some other assumptions about footing at this stage. In this case, using *DegenerateFoot to rule out the initial syllable of 'banana' requires the assumption that Prosodic words must begin with a foot (i.e. that Align-Ft-L is undominated) – but this does indeed seem to be the case for many early stages of English prosodic acqusition: see e.g. Kehoe (2000), and also §4.3 of this chapter.
[24] John McCarthy suggests that this stage could also arise if the learner assumes that the initially-stressed syllables are long and therefore do not constitute degenerate feet.

52) *the Greek intermediate stage:*
/wSw/ → [w(Sw)], *[(Sw)]
/wwSw/ → [w(Sw)], *[ww(Sw)]

53) *the data (taken from Revithiadou and Tzakosta (2004, ex. 4)* [25]

Subject	Target	Child Output	Gloss
B1 (2;09.25)	/ka.la.má.ci/	[ka:.(má.ci)]	'straw-diminutive'
B1 (2;09.12)	/ɣu.ru.ná.ca/	[ɣu.(ná.ca)]	'pigs-diminutive'
D (2;04.05)	/me.li.tí.ni/	[me.(tí.ni)]	(name)
D (2;04.05)	/fo.to.ɣra.fi.es/	[fa.(fi.eθ)]	'photographs'

As with Kehoe's banana vs. avocado data, this truncation pattern also requires the use of a faithfulness constraint relative to initial syllables, because what *markedness* constraint could be prompting the retention of an initial unstressed syllable? Thus, sandwiching a markedness constraint against unfooted syllables, Parse-σ, between positional and general faith again derives the right ranking for this intermediate stage:

54) Max[Seg]-σ1 >> Parse-σ >> Max[Seg]

55) Parse-σ rules out unfooted syllables… except those protected by Max(Seg)-σ_1

/ɣu.ru.ná.ca/	Max[Seg]-σ_1	Parse-σ	Max[Seg]
ɣu.ru.(ná.ca)		**!	
ɣu.(ná.ca)		*	*
[(ná.ca)]	*!		**

2.3.4 Faithfulness to morphological roots

The example of a Specific Faith stage relativized to roots that I present here is somewhat different from the rest in that it comes from historical sound change. The

[25] Admittedly, since these authors do not provide percentages of outputs that conform to each particular pattern, we do not know how much of a stage this really was. However, it seems encouraging at least that the two children I've used here used this truncation pattern on different words, at the same age.

relevant data, raised by Albright (to appear), come from a change from Middle High German (MHG) to Modern Northeast Yiddish, in which the process of final obstruent devoicing disappeared.

In Middle High German, final devoicing held in both roots and affixes, so that forms were either voiceless across the board, or alternating:

56) *Source language ranking (MHG):*
*FinalVcdObs >> Ident[vce]-Rt, Ident[vce]

In the case of roots, final consonants that had alternated in MHG between voiced and voiceless became uniformly *voiced* in Yiddish: compare the MHG nominative singulars and plurals in 57a) below with the corresponding Yiddish forms in 57b) (data from Albright, to appear examples 4 and 5) (data cited from Katz, 1987):

57) *MHG root-final voiced obstruents became **voiced** in Yiddish*

a) Middle High German forms[26]		b) Yiddish forms		glosses
nom. sing.	nom. plur.	sing.	plur.	
lo**p**	lobe	loy**b**	loyben	'praise'
ra**t**	reder	rɔ**d**	reder	'wheel'
ta**k**	tage	tɔ**g**	teg	'day'
hu**s**	hiuzer	hoy**z**	hoyzer	'house'
brie**f**	brieve	bri**v**	briv	'letter'

At the same time, Albright points out that the two Middle High German affixes that similarly alternated became uniformly *voiceless* in Yiddish:

[26] This may not be the correct phonetic transcription of the vowels for MHG forms, but the final obstruents in the singular forms are definitely correct.

58) *MHG affix-final voiced obstruents became **voiceless** in Yiddish*

a) MHG affix	b) Yiddish affix	c) Yiddish examples	gloss
[-ik, -i**g**e] (adjectival suffix)	[-ik, -i**k**e]	lebedik, lebedike, lebedikən, lebediker	'lively'
[ap, a**b**, a**b**e] (preposition/ prefix)	[ɔ**p**]	ɔpesn	'eat up'

Thus it would appear that in the transition between Middle High German and Yiddish, speakers re-ranked from the fully unmarked grammar in 56) to the Specific-F ranking in 59):

59) *Post-sound change ranking (Yiddish) – c.f. 56):*
Ident[vce]-Rt >> *FinalVcdObs >> Ident[vce]

The explanation for why root-final consonants all became *voiced* rather than voiceless is a somewhat separate matter; see Albright (to appear) for the argument that it was paradigmatic leveling to the plural, and for a somewhat different approach to this data in general. But as for this re-ranking of *FinalVcdObs, Albright says:

60) "The older stage of the language provided no evidence for the relative ranking of Ident-IOLexCat(voi)[27] and Ident-IO(voi) […] Therefore, we have no particular reason to expect that a demotion of the ban on voiced codas should have placed it below one constraint but not the other." (Albright, to appear: 9)

My goal is to suggest a mechanism whereby it *is* predicted that learners who are abandoning the fully M >> F ranking in 56) first move onto an intermediate ranking like 59). However – as Alright points out in his footnote 8: "If some external force managed

[27] His version of Ident[vce]-Rt

to create voiced obstruents just at the end of stems, but not affixes, [a ranking bias of Spec-F >> Gen-Faith] would indeed learn exactly the right grammar (attested in Modern Yiddish.) However, [that bias] does not straightforwardly explain why voiced codas should be created in the first place." It is certainly true that the extent to which my approach to gradual re-ranking can extend from synchronic developmental stages to diachronic sound change is not known, and will not be pursued further in this work.[28]

2.4 Summary of the data

The data presentation of this section has pushed a particular view of children's intermediate stages. Starting from the broad observation that children's grammars increase in markedness as they develop, I have suggested that the grammars of many such stages can be characterized with rankings in which a specific version of a constraint is crucially ranked above its more general counterpart(s), either markedness or faithfulness:

61) *Intermediate stages*
a) The Specific-M stage: Specific-M >> Faith >> General-M
b) The Specific-F stage: Specific-F >> Markedness >> General-F

With this view of the data, I now turn to the question of how a BCD learner might be reliably coerced into passing through such stages.

[28] The crucial obstacle for applying this dissertation's model in any discussion of sound change would be finding the *trigger* for re-ranking is, since this trigger cannot be the overt observational errors that child learners have at hand. Albright (to appear) contends that the relevant mechanism of sound change in this case was the learners' decision to use the plural as the paradigm's base; how this approach and my own might be integrated is a very interesting question for future research.

3. The theory of intermediate stages: Error-Selective BCD

As previewed in section 1, the heart of my proposal is that the learner should still use BCD as their re-ranking algorithm, in all its over-efficient splendour, but be conservative as to which errors it allows BCD to see when doing its re-ranking. Below I provide this mechanism, illustrate how it works using some of the examples we've already seen, and provide some discussion of its assumptions and workings.

A side note about an alternative theory before I begin: the OT phonological acquisition literature has seen much recent work using the Gradual Learning Algorithm (Boersma, 1997 *et seq.*), which is inherently designed to go through stages of acquisition. I alluded in chapter 1 to some problems that the GLA has in finding correct end-state grammars – all of these were to do with its choice not to retain errors for later reasoning, as the BCD does with its Support. Chapter 3 will turn to the GLA in earnest, and to the kinds of attested intermediate stages that it can (and cannot) derive.

3.1 The Error-Selective Learning proposal

Compared to straight BCD, my Error-Selective Learning approach (ESL) is different in two key respects: (a) what it does when it makes an error, and (b) what it does when it learns. With respect to the first, ESL retains the notion of the Support as the repository of errors that have been learned from, and which will be kept in mind each time re-ranking takes place. But ESL also uses another storage facility, called the Error Cache, which acts a holding pen for all the ERCs made on-line by the current grammar. Making an error does not trigger learning, but rather just an update of the Cache.

Periodically, the learner is triggered to stop merely accumulating errors and actually learn a new ranking. This triggering is done by a particular markedness constraint that has assigned Ls to ERC rows in the Cache – the details of how a constraint triggers learning are tied to input frequencies in a way discussed below. With respect to the second difference: learning proceeds in two steps. The first step is error selection: one particular error is chosen from the Cache to be added to the Support, and the second step is just re-ranking using the BCD algorithm already adopted.

3.1.1 What happens when an error is made: a Specific-M example

Error-Selective learning starts, as with normal BCD, with the making of errors and the building of ERC rows. But unlike the straightforward BCD approach: errors don't immediately appear in the Learning Support once they're made; nor do they trigger re-ranking. Instead, when an error is made its resulting ERC row is added to a temporary storage area, which I will call the Error Cache:

62) *How the Error-Selective BCD learner responds to making an error*

a) At an Early Stage: this error is made…

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

adding error to the Cache

b) … 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

c) … but the Learning Support Table is NOT updated:

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

And once the error has been added to the Cache: nothing else happens. While the learner has noted they've made an error, they do nothing immediate in response to that error, and so continue to use their current grammar, continue to make these (and other) errors and add them to the Cache, and the Cache keeps growing, until:

3.1.2 How learning is triggered

In Error-Selective learning, 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. I will refer to this particularly offending constraint as 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 three different winner~loser pairs in the Error Cache, learning is triggered:

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

Input	*Winner ~Loser*	***NoCoda***	**CompCoda*	*Max*	**CompOnset*
i) /frɛnd/	frɛnd ~ fɛ	**L**	L	W	L
ii) /piz/	piz ~ pi	**L**	e	W	e
iii) /qreᶦp/	qreᶦp ~ qe	**L**	L	W	L
iv) /ti/	ti ~ si	e	e	e	e

This Error Cache already shows the benefit of defining *ComplexCoda as a less stringent version of NoCoda. An L assigned by *ComplexCoda is always accompanied by an L from NoCoda, as in the errors 63i) and 63iii) above; whereas NoCoda can assign

Ls when *ComplexCoda is indifferent, as in 63ii). As a result, a more stringent markedness constraint will always reach its Violation Threshold before a less stringent one (much more on this point in section 4.)

3.1.3 Step 1: Choosing an error to learn from

Once learning has been triggered, the Error-Selective learner must choose one error from the cache – the Best Error – to learn from. As a first pass, we can choose errors via the two criteria I give below:

64) *The Error Selection Algorithm (ESA) (first pass)*
Choose as the best error that row in the Cache which:
a) has an L assigned by the Trigger Constraint,
and of those, the one that
b) has the fewest Ls assigned by *other* Markedness constraints,

I will return to the general consideration of why the ESA looks the way it does in section 3.2 below: for now, let us just apply these criteria to the Cache in 63) above. Criterion (a) eliminates 'tea', because it does not have a NoCoda violation. Criterion (b) eliminates 'friend' and 'grape', because they also have Ls assigned by *ComplexCoda and *CompOnset. Thus, the Best Error of these four is 63)ii):

65) *piz ~ pi is the chosen Best Error*

Input	*Winner ~Loser*	***NoCoda***	**CompCoda*	*Max*	**CompOnset*
i) /frɛnd/	frɛnd ~ fɛ	**L**	L	W	L
ii) /piz/	piz ~ pi	**L**	e	W	e
iii) /qreᶦp/	qreᶦp ~ qe	**L**	L	W	L
iv) /ti/	ti ~ si	e	e	W	e

After choosing the best error, the learner adds that error to the Support and empties the Cache. To illustrate how far the learner has gotten now:

66) *How the Error-Selective BCD Learner gets from errors to a new Support*

a) At an Early Stage: this error is made…

/frɛnd/	NoCoda	*Comp Coda	Max
fɛnd	*!	*	
fɛn	*!		*
☞ fɛ			**

error added to the Cache

b) NoCoda becomes the trigger constraint

Input	*Winner ~ Loser*	***No Coda***	**Comp Coda*	*Max*
/frɛnd/	frɛnd ~ fɛ	L	L	W
/piz/	piz ~ pi	L	e	W
/greˈp/	greˈp ~ ge	L	L	W
/ti/	ti ~ si	e	e	e

step I: ESA

d) Error Cache cleared…

Input	*Winner ~Loser*	*No Coda*	**Comp Coda*	*Max*
… empty, waiting …				

c) the pre-existing Learning Support Table

Input	Winner ~ Loser	No Coda	*Comp Coda	Max
… empty, waiting…				

e) … and Support updated

Input	Winner ~ Loser	No Coda	*Comp Coda	Max
/piz/	piz ~ pi	L	e	W

And now the learner moves on to step two of learning, which is simply:

3.1.4 Step 2: Applying BCD

In the case above, this means learning from the new Support piece of data in 66e) above. As we have seen a few times already, this ERC demonstrates the need to demote NoCoda below Max, but the e assigned by *Complex Coda means that the BCD learner can install this last constraint at the top of the hierarchy. Thus, ranking from this Support will get us to the ranking *ComplexCoda >> Max >> NoCoda – and this is the Specific-

M ranking we saw in section 1.2. This grammar protects the markedness of singleton codas, because the error 'peas' in the Support demonstrates that the target language tolerates them, but it still reduces complex codas to singletons.

3.1.5 A second example: a Specific-F stage

To see the role of Faithfulness in this error-selective decision-making process, I return to the French example of complex onsets in stressed vs. unstressed syllables. Recall that at the first stage of data, repeated below using Clara's data, all onset clusters are reduced:

67) Clara's stage 1 – all complex onsets reduced to singletons (1;0.28-1;09.01)

stressed syllables			*unstressed syllables*		
Target	Child	Gloss	Target	Child	Gloss
/kχa.ˈkχa/	[ka.ˈkæ]	'Cracra' (name)	/bʁiˈze/	[bœˈçiː]	'broken'
/plœʁ/	[pœː]	'(s/he) cries'	/abʁiˈko/	[pupæˈko]	'apricot'
/flœʁ/	[βœː]	'flower'			

The state of Clara's learning during stage one – just before stage two – is reflected in the Error Cache below:

68)

Input	*Winner ~Loser*	****Complex Onset***	*Max(Seg)*	*Max(Seg)-σ'*
i) /plœʁ/	ˈplœʁ ~ pœː	L	W	W
ii) /flœʁ/	ˈflœʁ ~ βœː	L	W	W
iii)/kχa.ˈkχa/	kχa.ˈkχa ~ ka.ˈkæ	L	W	W
iv) /bʁiˈze/	bʁiˈze ~ bœˈçiː	L	W	e
v) /abʁiˈko/	abʁiˈko ~ pupæˈko	L	W	e

This Error Cache has a number of errors with violations of *Complex – so, let us imagine we are at the stage where *Complex has triggered learning. Assuming that we have narrowed the candidates to these five using criterion (b) of the ESA, we now need a way of choosing among the errors in 68), and for that we need a third criterion:

69) *The Error Selection Algorithm (ESA) (full version)*

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

Criteria (c) tells us that of the errors in 68), we want an error that has the *most* Ws among the faithfulness constraints. Given this Cache, this will mean one that has Ws in both Max-Seg and Max-σ' columns, e.g. one of the first two:

70) *The best error(s) chosen*

Input	*Winner ~Loser*	***Complex Onset***	*Max(Seg)*	*Max(Seg)-σ'*
i) /plœʁ/	ˈplœʁ ~ pœː	**L**	W	W
ii) /flœʁ/	ˈflœʁ ~ βœː	**L**	W	W
iii)/kχa.ˈkχa/	kχa.ˈkχa ~ ka.ˈkæ	**L**	W	W
iv) /bʁiˈze/	bʁiˈze ~ bœˈçiː	**L**	W	e
v) /abʁiˈko/	abʁiˈko ~ pupæˈko	**L**	W	e

And adding one of these two ERCs to the Support will allow the specific >> general IO-faith bias to install just Max(Seg)-σ' above *ComplexOnset, giving us our intermediate stage:

71) *The Support*

Input	Winner ~ Loser	***Complex Onset***	Max-Seg	Max-Seg (σ')
i) /plœʁ/	ˈplœʁ ~ pœː	**L**	W	W

72) *Resulting BCD ranking*
Max[Seg]-σ' >> *ComplexOnset >> Max[Seg]

3.2 Discussion of the ESA, and Error-Selective Learning more generally

ESL is a way to learn from errors that will change the grammar as minimally as possible – while still being able to use the restrictive power of BCD. Since errors in the cache were all created by a current ranking, the ESA can find errors that only require small revisions to the present grammar (i.e. the demotion of a small number of L-preferring Markedness constraints), given the current lexicon.[29]

3.2.1 Analyzing the three ESA criteria for choosing errors

The Error-Selection Algorithm is the mechanism that decides which stages the learner goes through, because it chooses the errors that BCD builds its grammars from. The ESA's three criteria are thus built to ensure that the learner will choose errors that derive the kinds of intermediate stages discussed above. They are based on the logic of ERC rows – what their Ws and Ls that they contain tell us – and they are arranged in order of decreasing importance.

[29] As Elan Dresher points out (p.c.) the Error-Selective learner's reliance on the lexicon to provide minimal changes to the grammar is reminiscent of the Triggered Learning Algorithm (Gibson and Wexler, 1994). In this way, ESL is also similar, perhaps more so, to the learning model of Fodor (1998a,b) and subsequent – thanks to Lyn Frazier for discussion.

To understand the first two criteria: recall that the Markedness constraints that assign Ls to winner~loser pairs are those which are currently ranked too high in the current grammar; they are in some way responsible for the particular error that has been made.

The first criterion of the ESA is that chosen errors must have an L assigned by the Violation Threshold. This means that the learner will attends to the most frequent marked structure in the target that its current grammar does not allow – the constraint that triggered learning in the first place. The second criterion is that the chosen error has as few *other* Ls assigned by Markedness constraints. This means that the learner will choose an error that makes BCD demote as few other Markedness constraints as possible – in other words, that it will teach it as few new things as possible.

Together, these two criteria derive Specific-M stages, like the singleton vs. complex codas in §3.1.1-4 above. Once NoCoda has overcome the VT, the learner will add an error to the Support that shows that NoCoda must be demoted, but which says nothing about the ranking of as many other Markedness constraints as possible (like *ComplexCoda.) Given the BCD bias for high-ranking Markedness constraints, choosing an error that does not prove the need for demoting *ComplexCoda will allow it to stay at the top of the hierarchy.

To understand the third criterion: recall that among the faithfulness constraints that assign Ws in an ERC row, at least *one* of them must be ranked higher in the target grammar than in the current one. It is also important to realize the W-assigning properties of more vs. less stringent faithfulness constraints. A marked structure that is assigned an L by a less stringent faith constraint will also get one from the more stringent, general

faith constraint – e.g., if an onset is voiced in the target winner but devoiced in the loser, it will receive Ws from both Ident[vce]-Ons as well as general Ident[vce]. This in turn means that errors in which a marked structure appears in a privileged position will be assigned *more* Ws than if the same marked structure appears in a less privileged position – i.e., that compared to the voiced obstruent in onset that got two Ws assigned by Id[voice] constraints above, an obstruent in coda position that is devoiced in the loser will only garner a W from the general Id[voice] constraint.

The third criterion of the ESA is that chosen errors should have as *many* Ws assigned by faithfulness constraints as possible. As we've just seen, this criterion will choose errors that have marked structures in privileged contexts.

In conjunction with a ranking bias for specific >> general IO faith, this third criterion derives Specific-F stages; this was illustrated in the complex onset example of §3.1.5. Recall that BCD installs as few IO-faith constraints as it can and still resolve its errors, and that its biases ensure that it tries installing specific faithfulness constraints before general ones. So by choosing errors with Ws assigned by very specific faithfulness constraints, the learner ensures that BCD builds a grammar that allows marked structures only in those very specific contexts, and gradually learns their true scope as more errors are added to the Support.

3.2.2. Terminating ESL and converging on the end stage grammar[30]

How can we be sure that Error-Selective Learning will terminate? To do so, we must make sure that the final time an error is added to the Support, it is the ONLY remaining error – that is, that nothing unexplained remains in the Cache. With the system

[30] Thanks to John McCarthy for alerting me to this very necessary aspect of the proposal.

as it stands, it is unfortunately the case that some errors will *never* get added to the Support. If after some late stage of learning, the number of lexical items in the language on which the learner is still making errors *is less than the Violation Threshold,* then learning will never be triggered again but yet the learner will not have reached the end state. This is not a welcome result.

To make sure the Error-Selective learner terminates, we must make sure that eventually even a single error in the Cache can be added to the Support. Thus, I propose that the Violation Threshold is not just a fixed number, but rather a value that changes over time. The VT will begin fixed at its highest point, and decrease over time until it is eventually at one.[31]

With this caveat, the Error-Selective Learner will eventually end up with the same grammar as straight BCD would. Every time a constraint exceeds the Violation Threshold, some new error is chosen to add to the Support, and once BCD learns from an error, it is never made again. Because the Error Cache is emptied every time a new error is added to the Support – this will prevent the learner from being trapped in being triggered from very frequently violated constraints over and over again.[32] Eventually all the errors necessary to finding the correct grammar will be added to the Support, at which point no more errors are made and the learner will have reached the final state.

[31] Two remarks. First, it is might be fruitful in discussing language evolution to consider the effects of not letting the VT get as small as one, as a way to quantify how generalizations for which there is infrequent evidence (from perhaps only a few lexical items) are lost over generations of language acquisition. Second, note that the idea of a decreasing VT is related, but not directly analogous, to the decreasing re-ranking plasiticites of the GLA – see chapter 3.

[32] Although depending on the repairs that learner choose, Trigger Constraints may continue to be violated in later Error Caches. See section 3.3 below.

3.2.3 Irrelevant markedness violations

In choosing an error to move from Cache to Support as defined in the ESA above, the learner is taking into consideration *all* Markedness violations, and this may have somewhat unanticipated consequences. I illustrate this point in 73) below, with a slightly more articulated Error Cache involving NoCoda and *ComplexCoda. From this Cache, the learner will fail to go through the intermediate stage of singleton codas – just because the error with the singleton coda has a somewhat marked coda consonant, while another error has a complex coda with less marked segments:

73) *an Error Cache that triggers learning – but of a complex coda:*

Input	*Winner ~Loser*	***NoCoda***	**Comp Coda*	*Max*	**Comp Onset*	**Fricative*	**VcdObs*
i) /frɛnd/	frɛnd ~ fɛ	L	L	W	L	e	e
ii) /piz/	piz ~ pi	L	e	W	e	L	L
iii) /gre^{i}p/	gre^{i}p ~ ge	L	L	W	L	e	e
iv) /ti/	ti ~ si	e	e	e	e	e	e
v) /pant/	pant ~ pa	L	L	W	e	e	e

By counting the Ws assigned by markedness constraints other than NoCoda, we can see that criterion (b) of the Error-Selection Algorithm will choose the ERC row in 73v) with its one other L-prefering markedness constraint over any of the others, including 73ii) with its marked coda [z].

The upshot is that whether or not an ESL learner goes through the singleton coda stage depends on how marked the singleton vs. complex codas are in the particular errors in the Cache. So, while the learner will always be triggered to learn by NoCoda before *ComplexCoda, they are *not* guaranteed to choose a best error that will push them

through this stage. Overall, this seems like the right prediction: e.g. the onset sonority effects that Trevor and Julia display are not true of all learners. With respect to stages of specific faithfulness, recall the conflicting results of Rose (2000) and Kehoe and Debove-Hilaire (2003) as to which French onset clusters give rise to the intermediate stage of stressed-syllable faith only in different children. One could complicate the learner further to ensure the stringency result – e.g. by adding more analysis of the other M violations – but it is not clear that the current, simpler method is necessarily undesirable.[33]

3.2.4 Choosing among positional faithfulness contexts

One aspect of criterion (c) – which favours errors with as many faithfulness Ws as possible – comes from French complex onsets example. With the right set of faithfulness constraints, triggering the ESA with the markedness constraint *ComplexOnset should lead the learner to pick an error that has a complex onset in a monosyllabic word. In a one syllable word, the syllable with the Markedness violation will be in both the stressed *and initial* syllables, and this will result in the most faithfulness Ws:

74) *A French learner's Error Cache, repeated*

Input	*Winner ~Loser*	***Complex Onset***	*Max-Seg*	*Max-Seg (Stressedσ)*	*Max-Seg (σ1)*
i) /plœʁ/	ˈplœʁ ~ pœː	**L**	W	W	W
ii) /flœʁ/	ˈflœʁ ~ βœː	**L**	W	W	W
iii)/kχa.ˈkχa/	kχa.ˈkχa ~ ka.ˈkæ	**L**	W	W	e
iv) /bʁiˈze/	bʁiˈze ~ bœˈçiː	**L**	W	e	W
v) /abʁiˈko/	abʁiˈko ~ pupæˈko	**L**	W	e	e

[33] John McCarthy (p.c.) points out that Error Caches like 73) will not arise only if two conditions are met: (i) the Violation Threshold is (initially) set sufficiently high to give the learner a representative sample of errors, and (ii) languages are assumed to be harmonically complete (on this notion, see esp. Smolensky and Legendre (2006) chapter 14.)

It will then be up to BCD to decide which faithfulness constraint to install above *ComplexOnset; let us assume that this French learner has already established via an illustrative Context Table from chapter 1 that French's initial syllables and stressed syllables are in no subset relationship. In the absence of any other relevant data, the learner may well choose to install Max(Seg)-σ_1 over *ComplexOnset. While this has no deleterious consequences for the end-state grammar, it does predict that French-learning children could go through a stage where onsets are retained only in initial syllables, and nowhere else.

3.2.5 The Violation Threshold and extra-grammatical factors

In the Error-Selective model, Violation Thresholds provide the interface between the language-specific module, with its knowledge of phonological constraints and abstract representations, and all more general cognitive factors in language development.

As already mentioned in the previous section, learners' VTs must decrease over time to ensure that they can eventually get all necessary errors in the Support to finish learning. The *rate* at which the VT decreases will determine the speed with which intermediate stages are overcome and new grammars are learned – and the correct initial values and rate of decrease are empirical questions (which this dissertation has far too little data to answer.) But it seems plausible that both of these parameters would vary from child to child as a function of individual cognitive abilities, and from context to context as a function of all other current cognitive demands on a child.

Allowing for different Violation Threshold values at different moments in learning also opens the ESL to one possible treatment of variability between rankings

over the course of development -- and even perhaps regressions, where learners temporarily return to an earlier stage after having mastered a later one. These possibilities will be the focus of section 5.

3.3 Illustrating ESL: a case study of Trevor and Julia's onset clusters

This section dissects Trevor and Julia's stop-initial onset cluster acquisition from §2.2.2 in more detail, with the ESL proposal in mind. I present a few stages along the way to complete mastery of onset clusters, and demonstrate how the Error-Selective learner can get from each stage to the next. Note that I follow each child up until the point where their data becomes insufficiently transcribed, meaning that that they are both not fully finished cluster acquisition at their final stages here.

I focus here just on stop-initial and s-initial clusters, because they are sufficiently attested in the data to make what I think are confident generalizations. While the sonority constraints that I adopt make predictions about the concurrent acquisition of other fricative-initial onset clusters, I leave them out of the current analyses.[34]

3.3.1 Trevor

This section will focus on Trevor's first three stages of onset cluster acquisition. At his first stage, which lasts until approximately 2;2, all clusters are reduced to singleton onsets only. At his second stage, which lasts about two months from 2;3-2;4, he permits

[34] It should perhaps also be noted, however, that the sonority difference between fricatives and stops may not in fact be relevant to the typology of permissible onset clusters: see Morelli (1999). If this were true, then the sonority hierarchy relevant to building the set of Onset Sonority Constraints in 27) would be collapsed to contain as its less sonorous element 'obstruents' – and this would make different predictions about the stages discussed here.

stop-r and stop-w clusters[35] but continues to reduce stop-liquid and all s-initial clusters to singletons. His third stage begins around 2;5, when he adds stop-liquid clusters to his inventory, but still reduces s-initial ones. To summarize, then (using capital S for stops):

75) Trevor's onsets at the first three stages

Stage 1:	[CV…],	*[CCV…]
Stage 2:	[SrV…], [SwV…], [SV…]	*[SlV…], [sCV…]
Stage 3:	[SlV…], [SrV…], [SwV…], [CV…]	*[sCV…]

An important caveat: these stages are in fact abstractions from the quantitative patterns of cluster preservation and reduction that Trevor passes through. The numbers I provide below will show that these are the *prevalent* productions at each stage, but clusters of each type are, in fact, reduced and retained to varying degrees throughout this period. For now I put aside the treatment of this variation, and discuss the ESL route through Trevor's stages as though they were all categorical, with the promissory note that section 5 will deal with some of these variation issues.

Getting from stage 1 to stage 2:

We begin at stage one:

[35] Trevor has only one stop-j cluster – 'piano' – which is consistently reduced. Perhaps Trevor is not perceiving this glide, or perhaps independent markedness constraints are ruling out [pj].

76) The end of Trevor's stage 1: all clusters reduced (up to 2;2)

		raw #				*percentages*			
age	*output*	*stop-l*	*stop-r*	*tr*	*sC*	*stop-l*	*stop-r*	*tr*	*sC*
2;0	C…	31	31	32	15	**100.0**	**86.1**	**82.1**	**100.0**
	CC…		5	7		0.0	14.7	23.3	0.0
2;1	C…	65	29	23		**94.2**	**69.0**	**76.7**	
	CC…	4	13	7	3[36]	5.8	31.0	23.3	
2;2	C…	30	34	*19*	32	**100.0**	**82.9**	50.0	**100.0**
	CC…		7	*19*		0.0	17.1	50.0[37]	0.0

The relevant fragment of Trevor's grammar at this stage is fully M >> F: all of the Onset Sonority constraints (*TW, and everything else) rank above faithfulness, e.g.:

77) *TW, *TR, *TL …. >> Max

Based on the words in 69) above, Trevor's Error Cache at 2;2 includes errors as in 78) below:

78) *A fragment of Trevor's Error Cache at the end of stage 1*

Word	*Winner ~Loser*	***TW***	**TR*	**TL*	*Other Mkdness*	*Max*
'blocks'	blɑks ~ gak	L	L	L	?	W
'glasses'	glæsɪz ~ gæʃij	L	L	L	?	W
'clock'	klɑk ~ kak	L	L	L	?	W
'cracker'	krækɪ ~ kaka	L	L	e	?	W
'train'	treɪn ~ teɪn	L	L	e	?	W
'between'[38]	bəˈtwɪn ~ tiː	L	e	e	?	W

[36] I consider these three tokens of /sC/ productions to be an aberration.
[37] As mentioned in section 2, Trevor's acquisition of [tr] is different from the rest of his obstruent-r clusters.
[38] This error is not from the corpus, but is rather inferred – as the footnote below points out, Trevor had so few stop-w inputs that we can't be sure when he started allowing them. Whether it was at 2;2 or earlier, he would have initially not tolerated them and so made errors such as this one.

To trigger learning, one of these markedness constraints must overcome the Violation Threshold. Since these markedness constraints are in a stringency relation, the first constraint to do so will be the most stringent, i.e. *TW (already shaded in the Cache above – see below on why I've shaded *TR as well))

Once *TW has triggered learning, Trevor now must search the Cache to find the best error, which must at least (a) violate *TW and (b) violate as few other M constraints as possible. Since Trevor's second stage permits both stop-w and stop-r clusters, one possibility is that among the errors violating *TW, the best one also violated *TR – that is, that those few errors with stop-w clusters happened to have more *other* Markedness Ls than the Cr ones. A second possibility is that between what I have called stages 1 and 2, Trevor learns twice in quick succession: triggered first by *TW, and then triggered again by *TR almost immediately afterwards. This is not so implausible given that Trevor has very few words with stop-w clusters compared to stop-r words – in fact, I have not given numerical data from Trevor's stop-w clusters above precisely because they are so infrequent.[39] This suggests that *TR would have reached its threshold soon after *TW. (Drawing the line between the activitiy of these two constraints is in any case difficult given that at this stage /r/ is frequently mapped to [w]).

In either case: once the learning dust settles, Trevor's Error Cache has been cleared, and his Support has been updated with an error containing a Cr cluster. From this Cache, I have chosen 'train':

[39] In the entire corpus up until 2;2, his only two attempted stop-w clusters are [**tw**iː] 'between' (2;1.14), and [**sk**ɪz] 'squeeze' (2;1.14).

79) *Trevor's Support after step 1 of learning:*

Input	Winner ~Loser	*TW	*TR	*TL	Max
'train'	treɪn ~ teɪn	L	L	e	W

And after step 2 – applying BCD to this Support – Trevor gets the ranking in 78) below that protects Cr and Cw clusters:

80) *Stage 2 ranking, from the Support in 79)*
*TL, *… >> Max[Seg] >> <u>*TW, *TR</u>
newly demoted

Getting from stage 2 to stage 3:

At stage 2, Trevor's ranking allows stop-r onsets, but still reduces most other onset clusters:

81) Trevor's stage 2 (numbers by output):

		raw #				*percentages*			
age	*output*	*stop-l*	*stop-r*	*tr*	*sC*	*stop-l*	*stop-r*	*tr*	*sC*
2;3	C…	23	12	*35*	17	**69.7**	31.6	89.7	**77.3**
	CC…	10	26	*4*	6	30.3	**68.4**	10.3	22.7
2;4	C…	28	18	25	16	**62.2**	41.9	53.2	**84.2**
	CC…	17	25	22	3	37.8	**58.1**	46.8	15.8
2;5	C…	14	4	5	21	35.9	20.0	22.7	**72.4**
	CC…	25	16	17	8	**64.1**	**80.0**	77.3	27.6

After a couple months of this, Trevor is again triggered to learn by an error in his Cache. Since his Cache was cleared when he added an error to the Support to get to stage 2, none of his errors on stop-w and stop-r clusters remain there. In the present errors (see the Cache in 82 below), *TL prefers the losers in his onset cluster errors – but so do *TW

and *TR. Thus, all *three* of these constraints will overcome the VT and trigger learning at the same time – I have only chosen *TL to shade as the Trigger Constraint below because it is this constraint whose position will be changed in the eventual re-ranking:

82) *A fragment of Trevor's Error Cache at 2;5*[40]

Word	*Winner ~Loser*	**TW*	**TR*	****TL***	**SN*	**ST*	*Other Mkdness*	*Max*
'glass'	glæs ~ gæs	L	L	L	e	e	?	W
'play'	pleɪ ~ peɪ	L	L	L	e	e	?	W
'cleaner'	klɪnɪ ~ kiːnə	L	L	L	e	e	?	W
'sneakers'	snɪkɪz ~ ənɪkəθ	L	L	L	L	e	?	W
'stick'	stɪk ~ dɪk	L	L	L	L	L	?	W

An important question about this stage is why, if *TW and *TR have been demoted, clusters like TL are still being deleted rather than mapped to better sonority clusters: in other words, why does 'glass' come out as 'gas' and not 'gwas' or 'gras'? The explanation will have to come from the ranking of other constraints. As we've seen, deletion (e.g. /gl/ → [g]) violates Max; a featural mutation like (/gl/ → [gw]) violates Ident constraints, as well (possibly) as other markedness constraints; some of these latter constraints must be currently ranked above Max in Trevor's grammar.[41]

Getting back to the Cache in 82) above): Trevor will now choose a Best Error that violates *TL but no *more* onset cluster constraints. Supposing that he chooses his error on 'glass', he adds it to the Support, and applies BCD to find the new ranking in 84):

[40] Beyond the data cited earlier – the new errors, 'sneakers' and 'stick' are from 2;4.3.
[41] Beyond these facts about Trevor, it is more generally the case that children's grammars often prefer deletion over other repairs. I cannot fully treat this issue here, but it is an interesting and outstanding question to what extent the deletion preference can be reduced to the activity of other constraints or requires e.g. some initial rankings among faithfulness constraints.

83) *Trevor's new Support:*

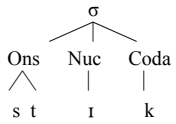
Input	Winner ~Loser	*TW	*TR	*TL	*SN	*ST	Max
'train'	treʲn ~ teːn	L	L	e	e	e	W
'glass'	glæs ~ gæs	L	L	L	e	e	W

84) *Stage 3 ranking, from the support in 81):*
...*SN, *ST … >> Max >> *TW, *TR, *TL
newly demoted

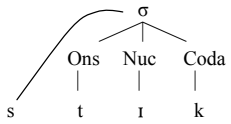
At this point, the major class of onset clusters that Trevor has not learned is the s-initial cluster, and the best treatment of this lag depends on assumptions about Trevor's syllabification of those inputs with sC sequences. The issue is whether the child is treating them as true sC onset clusters, or as singleton C onsets with the s in an adjunct position outside the syllable. (In the developing English context, see e.g. Barlow 2001, Goad and Rose, 2004; Chambless, 2006):

85) *Two possible syllabifications for 'stick'*

a) *complex onset*



b) *s-appendix plus singleton onset*



Trevor's consistent pattern of /sC/ → [C] reduction, regardless of relative sonority, allows two possible interpretations. The assumption I've been making above so far is that he is treating them as clusters, and so they violate so many sonority sequencing constraints that the ESA will not yet have chosen to add them to the Support (recall that the ESA's criterion (b) wants errors that have as few L-preferring Markedness constraints

as possible, ignoring the Trigger Constraint.) The alternative hypothesis is that they are adjuncts and the *Appendix constraint has yet to be demoted, as below.

86) *An alternative version of the Error Cache at 2;5*

Word	*Winner ~Loser*	**TW*	**TR*	****TL***	**Appendix*	*Other Mkdness*	*Max*
'glass'	glæs ~ gæs	L	L	L	e	?	W
'play'	pleʲ ~ peʲ	L	L	L	e	?	W
'cleaner'	klɪnɪ ~ kiːnə	L	L	L	e	?	W
'sneakers'	snɪkɪz ~ ənɪkəθ	e	e	e	L	?	W
'stick'	stɪk ~ dɪk	e	e	e	L	?	W

3.3.2 Julia

Julia's acquisition of onset clusters is a little more complicated than Trevor's, because she treats s-initial clusters differently depending on their sonority profile. But her first two stages are just like Trevor's (see below), so I pick up her ESL story at stage 2:

87) *Stage 1*: singleton onsets only (up to 1;9)
Stage 2: stop-r and stop-w onsets only (during 1;10)

Starting with stage 2:

As with Trevor, Julia's stage 2 reduces permits few s-initial onset clusters of any sort:

88) Julia's stage 2 (numbers by output):

age	*output*	*raw #s*			*percentages*		
		stop-r/w	*stop-l*	*sC*	*stop-r/w*	*stop-r*	*sC*
1;10	C…	9	21	10	26.5	**95.5**	**80**
	CC…	25	1	2	**73.5**	4.5	20

89) Representative data from Stage 2

clusters retained: stop-r, stop-glide			*clusters reduced: stop-l, s-initial*		
Target	*Child*	*Age*	*Target*	*Child*	*Age*
'drink'	[gwɪŋk]	1;10.5	'spoon	[pun]	1;10.8
'Grundy'	[gwʌni]	1;10.5	'sleep'	[sip]	1;10.7
'crackers'	[kwækəs]	1;10.14	'glasses'	[g̲aθəs]	1;10.10
			'please'	[p̲ɪs]	1;10.10

Thus Julia's Support at stage 2 is equivalent to Trevor's – that is, it contains errors that demonstrate the need to demote *TW and *TR.

90) Julia's Support at stage 2:

Input	Winner ~Loser	*TW	*TR	*TL	*SN	*ST	Max
'cry'	krai~ kai	L	L	e	e	e	W

Getting from stage 2 to 3

Julia's stage 3 of learning, at around 1;11-2;0, adds s-stop and s-nasal clusters to her onset inventory – while stop-liquid, s-liquid and s-glide clusters continue to be reduced. The table and data below illustrate this:

91) Stage 3: s+[-cont] (s-stop and s-nasal) onsets also appear at 1;11-2;0

age	*output*	*raw #s*				*percentages*			
		stop-r/w	*stop-l*	*s-stop, s-nasal*	*sl, sw*	*stop-r/w*	*stop-l*	*s-stop, s-nasal*	*sl, sw*
1;11	C…	22	1	4	5	**100.0**	**97.6**	14.3	**80**
	CC…	0	40	24	1	0.0	2.4	**85.7**	20
2;0	C…	23	3	1	7	**100.0**	**92.1**	3	**87.5**
	CC…	0	35	32	1	0.0	7.9	**97**	12.5

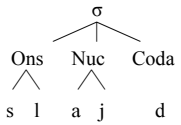
92) Representative data from Julia's stage 3:

clusters retained: stop-w, stop-r, s-stop, s-nasal			*clusters reduced: stop-l, sl, sw*		
Target	*Child*	*Age*	*Target*	*Child*	*Age*
'queen'	[gwɪn]	2;0.2	'clap'	[kæp]	1;11.15
'crown'	[kwaʊn]	2;0.2	'slipper'	[sipə]	2;0.18
'spilled'	[spɪod]	1;11.22	'slide'	[sai:t]	1;11.16
'sneeze'	[snis]	1;11.26	'swim'	[fɪm]	1;11.15

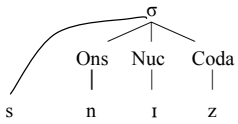
To get to this stage 3, Julia must choose a best error that will let her learn s[-cont] but not s[+cont] onsets. But what is the triggering constraint? It clearly cannot be onset sonority that prefers ST and SN clusters (look at the violations in the previous Error Cache.) Instead, I suggest that the right triggering constraint is *Appendix, and that the s-initial clusters that Julia learns at stage 3 are those that her grammar has parsed as containing not a complex onset but as a singleton onset and an s-appendix (see ranking below):

93) *Julia's sC syllabifications*

a) sl, sw: *complex onset*



b) s-stop, s-nasal: *appendix plus singleton*



Therefore, I adopt the two syllabification patterns in 93), to illustrate Julia's Error Cache below.[42] With this assumption, we can see that if *Appendix triggers learning Julia will choose a best error with an s-[-cont] onset cluster:

94) *Julia's Error Cache at the end of 1;10*

Word	*Winner ~Loser*	**TW, *TR*	**TL*	**SL*	****Appendix***	*Other Mkdness*	*Max*
'please'	pliz ~ piz	L	L	e	e	?	W
'sleep'	slip ~ sip	L	L	L	e	?	W
'spoon'	sp^{h}un ~ pun	e	e	e	L	?	W
'stairs'	sterz ~ deəz	e	e	e	L	?	W
'snake'	snek ~ nek	e	e	e	L	?	W

If for example Julia chooses 'spoon', her Support is duly updated as in 95), she then applies BCD, and she thus gets the stage 3 ranking:

95) *Julia's new Support*

Input	Winner ~Loser	*TW, *TR	*TL	*SL	*SN	*Appendix	Max
'cry'	kra^{j} ~ kai	L	e	e	e	e	W
'spoon'	s.p^{h}un ~ pun	e	e	e	e	L	W

96) *The new ranking at stage 3, from the Support in 93)*
*TL…*SN, *ST… >> Max >> *TW, *TR, <u>*Appendix</u>
newly demoted

[42] This split in her syllabification might be linked to her choice of segments when these clusters are reduced. As we have seen here, Julia reduces sl and sw clusters to [s], but s-stop and s-nasal clusters to the stop or nasal. If we assume that onset selection is driven by a preference for the least sonorous onset segment (with respect to children, see e.g. Gnanadesikan, 1995/2004; Pater and Barlow, 2003; Goad and Rose, 2004; cf. van der Pas, 2004), then it would be surprising that she chose to reduce complex onset /sn/ clusters to just the more sonorous nasal. If, however, that cluster is syllabified with only the nasal in onset, the pattern is explained.

An important point here is why SW and SL clusters don't surface faithfully at stage 3 by being parsed in the output as appendices. The answer must lie in the ranking of other markedness constraints: the likely candidates are syllable contact constraints. On the assumption that the appendix-onset boundary is assessed by such constraints, high-ranking constraints that prohibit too sharp a rise in sonority across the syllable boundary will rule out the appendix parse for SW and SL clusters (on syllable contact, see Murray and Venneman, 1983; Clements, 1990; on an OT analysis in the current system's spirit, see Gouskova, 2004):

97) *Julia's grammar chooses reduction of SL and SW*

'sleep' /slip/	Syllable Contact	*SL	Max-[Seg]	*Appendix
s.lip	*!			*
slip		*!		
☞ sip			*	*

3.3.3 Summary

This section has considered several stages in the acquisition of onset clusters by two different children. I have shown in ESL how the stringency of onset sonority constraints can predict well-attested stages, whereby better onset clusters are acquired before less good ones. I have also relied on s-initial clusters' variable syllabification to explain the differences between Trevor and Julia's development.

One interesting point of comparison between these two children is that Julia and Trevor go through the same kinds of onset cluster stages (modulo the sC differences), but Julia's are much *earlier* than Trevor's. Why should this be? In ESL there are two options: either she has more errors earlier, or she has lower Violation Thresholds. Both of these

options seem plausible, and while in the present system this is a mechanical, rather than an empirical issue, some important questions arise from the consideration of these mechanics.

The second option makes sense in the terms of §3.2.4 above, with which we can consider VTs as flexible thresholds of a rather psychological nature: affected by other cognitive demands and individual abilities, decreasing over time as language processing gets easier for the learner[43], and the like.

The first option – that Julia's Error Caches grow faster than Trevor's – raises the somewhat unanswered question of how precisely errors get made and into the Cache. The next section deals in part with this question, because it considers how frequency affects Error-Selective Learning. As we will see, what the empirical predictions connect are ambient lexical frequencies and constraint stringency on the one hand and *order* of acquisition on the other. However, they have nothing central to say about *rates* of acquisition. In any event, the difference between Julia and Trevor's rate of onset cluster learning will remain outside the predictive domain of this fundamentally grammatical, not psychological, approach to acquisition.

4 The roles of frequency

4.1 The connection between frequency and Error-Selective Learning

There has been extensive discussion in the literature of how lexical frequency influences acquisition order. Drawing the right formal connections between frequency and grammar is clearly a long-standing point of controversy, no less tricky in the domain of acquisition than in phonological theory as a whole. However, an important fact that the

[43] Similar to the decreasing plasticity of the GLA – see chapter 3 section 1.

Error-Selective approach exploits is that incorporating frequencies or statistics across the lexicon into *learning* is logically independent from using those frequencies or statistics directly in the *grammar* – either in the definition of constraints or in the workings of EVAL.[44]

In Error-Selective Learning, the ideas of Violation Thresholds and the ESA criterion (b) that favours errors with the fewest *other* Ls, together conspire to predict that order of acquisition should mirror markedness violation frequency. The more errors that a constraint assigns Ls to, the earlier one of those L-assigned errors will get into the Cache, and so the earlier it will be demoted.[45]

In these two ways, the ESA only makes universal predictions about order of acquisition among more vs. less stringent M and F constraints (and then not even a completely deterministic prediction, as pointed out in §3.2.2). The relative re-ranking of constraints *not* in stringency relations, on the other hand, is in no way fixed beforehand. Instead, these ordering decisions will be specific to the target language, and the particular errors the learner has added to their Cache.

4.2 The connection between frequency and order of acquisition

The arguments in the literature connecting frequency and stages involve cross-linguistic comparisons, in both absolute and relative terms (for a recent brief review, see Beckman and Edwards, 2000). Here I discuss one robust example, from a series of studies that together demonstrate how differences in order of acquisition between

[44] Thanks to Sharon Goldwater for discussion of this point.
[45] On the Faithfulness side, the ESA criterion (c) that favours errors with the most Ws predicts something comparable but not hinged on frequency – that the more privileged positions a marked structure appears in, the earlier an error that forces its acquisition will be added to the Cache.

Germanic (German, English, Dutch) and Romance (Spanish, at least) are the result of language-specific input frequencies.

4.2.1 Data from cross-linguistic frequency: initial weak syllables vs. codas

The distillation of this cross-linguistic comparison comes from Roark and Demuth (2000). The background for their study are the two generalizations given below:

98) *Two generalizations about cross-linguistic order of acquisition*

a) Initial unstressed syllables appear in Spanish before Germanic
(Lléo 1997,1998; Lléo and Demuth, 1999; Demuth, 2001)

b) Coda consonants appear in Germanic before Spanish
(Lléo et al, 1996; Lléo and Prinz, 1997; Lléo and Demuth, 1999)

In Spanish, initial weak syllables begin to surface somewhere between 1;6 and 1;10. For example, Lléo (2003) finds that roughly 40% of the utterances from her two Spanish-learning children at 1;6 already contain initial unstressed syllables. In contrast, English unstressed initial syllables appear somewhere shortly after 2. Trevor and Julia both begin to produce them at around 2;0; Gerken (1994) reports them appearing as late as 2 and a half. Meanwhile, Spanish codas are learned starting around 1;10 at the earliest; as seen in Lléo (2003)'s data in §2.3.1, José doesn't get them before 2;0 (see also Lléo 1997; Gennari and Demuth, 1997). In English, however, codas usually appear before or at the middle of the second year – for example, both Trevor and Julia acquire singleton codas between 1;4 and 1;6. It has also been reported for both German (Grijzenhout and Joppen, 1998) and English (Salidis and Johnson, 1997; Velleman and Vihman, 2000, 2002b) that some children's very first productions contain codas. In addition, Lléo et al

(1996) report that the proportion of closed syllables is already significantly higher in Gerrman than Spanish at the 25 word mark.[46]

With these findings as background, Roark and Demuth (2000) compare data from Spanish and English to demonstrate that these structures are highly frequent in the language in which they appear first: initial weak syllables in Spanish, and codas in English. Their data came from corpora of child-directed speech in the two languages, using two lexicons of 18,000 words each extracted from CHILDES (thus admittedly reflecting token, not type, frequencies.)

First, they found that English initial syllables were stressed in an overwhelming portion of the corpus – monosyllabic words and disyllabic trochees already accounted for about 90 percent of the English tokens, (even when they allowed for the possibilities of encliticized 'the' and 'a'). In Spanish, however, 40% of the data contained initial unstressed syllables.[47] Furthermore, of the other 60% which had stressed initial syllables, more than a third of tokens (26% of the total) were due to just 10 extremely common words, mostly functional items: *con, en, es, no, por, qúe, sí, ver, y,* and *ya*. In contrast, 59.3% of the English words in that same sample had coda consonants, while only 25.2% of the Spanish words had codas.

A related study by Kirk and Demuth (2003) also found that the prevalent tendency of English children to learn complex codas before onsets (which was also true of Trevor) correlates with the frequency of these two structures in child-directed speech, rather than any of the other possible predictors they consider.

[46] Shelley Velleman (p.c.) reminds me that these differences also appear in babble.
[47] Of these 40% -- 10% were wS words, almost 20% were wSw, and the remaining 10% were longer words with initial w.

The overall finding here is that the frequency with which (at least some) marked structures occur in children's inputs correlates closely with their relative order of acquisition. In the Error-Selective Learning approach this connection is predicted, because markedness constraints that are violated frequently in the input will frequently create errors and so reach their Violation Thresholds earlier than infrequent ones.

A corollary of this connection is that structures with equal frequency in the child-directed lexicon should be variable in their order of acquisition. This prediction is discussed explicitly in the work and interpretation of Levelt and van der Vijver (2004), which discusses the order of acquisition of complex onsets vs. codas in the Fikkert/Levelt corpus. Among the 12 children acquiring Dutch in that sample, 3 acquire CCVC syllables before CVCC, while 9 acquire CVCC before CCVC. Since the frequency of both these syllable types is comparable in the child-directed Dutch they report on (3.4% of their Dutch corpus being CCVC vs. 3.7% being CVCC), the ESL explanation of this variation is the particular frequency quirks of the Error Cache that each child builds, based on their individual lexicons and experience.

4.2.2 Ambient not output frequencies, and the Error Cache

What we have seen above is that the lexical (token or type) frequencies in child-directed speech seem to be the driving factor. In the ESL theory of development, it is worth considering how those frequencies get mirrored in the errors that make it into the Cache.

The simplest assumption would be that children produce words with about the same frequencies as they hear them, and that the Error Cache is populated by all and only

children's overt production errors.[48] If the child's own production error frequencies were the right predictor, then if a child's lexicon were skewed – e.g. his words included a great many complex onsets and only a few complex codas – they would be predicted to acquire the former before the latter.

But looking at Trevor and Julia's outputs demonstrates for certain cases that *ambient* input frequencies are what drive the triggering process, and *not* a particular child's production error frequencies. The clear counter-example: Trevor uses more words with complex onsets than complex codas (and by token, MANY more – see table 97). Nevertheless, he acquires complex codas around 1;8-1;9, whereas complex onsets do not appear nearly at all in his productions until 2;2 (and not reliably until about 2;4), as in table 100).

99) *Complex syllable margins in the **targets** – Trevor*

age	*by word token (during the stage)*		age: up to…	*by word type (totals)*	
	CompOnset	CompCoda		CompOnset	CompCoda
0;10-1;3	**101**	15	1;3	11	7
1;4	**51**	17	1;4	13	8
1;5	**94**	41	1;5	24	15
1;6	59	54	1;6	28	20
1;7	96	90	1;7	38	32
1;8	116	82	1;8	53	45
total	517	299			

[48] Related to this line is the argument in Fikkert and Levelt (to appear) that the prevalence of certain places of articulation in a child's early lexicon dictates their patterns of consonant harmony.

100) *Complex syllable margins in the **outputs** - Trevor*

	Complex onset inputs			*Complex coda inputs*			
	C..	CC…	% CC	…0	…C	…CC	% CC
up to 1;3	101	0	0.0	11	3	1	6.7
1;5	93	1	1.1	10	25	5[49]	12.5
1;6	59	0	0.0	4	29	14	29.8
1;7	91	2	2.2	12	46	29	33.3
1;8	104	11	9.6	6	29	**45**	**56.3**
1;92	100	9	8.3	5	21	**89**	**77.4**
1;10	154	15	8.9	3	34	**170**	**82.1**
1;11	86	20	18.9	4	14	**93**	**83.8**
2;0	127	14	9.9	6	43	**66**	**57.4**
2;1	149	31	17.2	5	57	**110**	**64.0**
2;2	118 (19)[50]	30	20.3	2	49	**112**	**68.7**
2;3	87 (31)	46	34.6				
2;4	74 (25)	**80**	**51.9**				
2;5	31 (5)	**78**	**71.6**				

The most immediate consequence of this point touches on the nature of my proposed Error Cache, since it is the locus of frequency effects in ESL. What this data suggests is that we must understand the errors in the Cache to include not only overt production errors made by the learning child, but crucially also errors resulting from passive listening. One potential source of these errors might be the early application of a perception grammar of the sort proposed in Pater (2004) or Boersma (2001). The general idea is that perception errors are created when the child correctly hears a target word at some auditory level, but feeds that form as an input to their current perception grammar, and then notices that their perceptual grammar has mapped the phonetic form

[49] All 5 of these are productions of the word 'bump'.

[50] Of the total reduced clusters, the number in brackets are the reduction of 'tr' in 'Trevor', which around 2;2 he began to pronounce more than half the time with an initial [t] or [tʃ].

unfaithfully. From this noticing would come a silent ERC row – normal in all respects except its lack of phonetic implementation – which the learner will add to their Cache.[51]

Including such passive errors in the Cache also allows us to interpret the fact that at the very first stages of production, some children have clearly demoted some markedness constraints below conflicting faithfulness constraints (like NoCoda in English and German, as cited in section 4.2.1.[52]) While children may never have produced words with codas before – they have surely heard many many words with them, and had time to build up enough errors to trigger learning on NoCoda. (Alternatively, the necessary errors might indeed have come from production, in late stages of canonical babble.)

However: this notion of including *perception* knowledge in the mechanism used for learning a *production* grammar clearly raises deeper questions about how these two kinds of phonological knowledge interact in development – questions that I take to be fundamentally unanswered (though see especially Pater 2004, Pater, Stager and Werker 2004, and also Escudero and Boersma 2003.) Research over the last decades has shown that children are sensitive enough to the frequencies in their linguistic input to prefer more frequent structures very early in life (see e.g. Jusczyk, Frederici et al (1993); Jusczyk, Luce and Charles-Luce (1994); Jusczyk (1997) and references therein.) Clearly, however, this sensitivity and awareness does not build the learner a fully-target grammar

[51] One important consideration is that depending on the theory, the kinds of faithfulness constraints used in the perceptual parsing grammar may be different from those used in creating production errors – this is definitely the case in Boersma (2001) and to a lesser extent in Pater (2004) – so the kinds of ERCs rows produced in early perception will not necessarily mirror attested early errors in production. The spelling-out of this consideration and its consequences will be left unresolved here.

[52] With respect to this early F >> M ranking, see also the different approach to stages of acquisition in Bernhardt and Stemberger (1998).

to be used when production begins some months later – but yet the same frequencies are again relevant to that grammar's development as well.

4.3 Intermediate stages without stringency: stages of prosodic truncation

The previous sections on frequency have pointed out that ESL does not make universal predictions about the order of demotion of constraints not in stringency relations – but also that it does make general predictions about the stages and frequencies in the input. In this section I demonstrate this second point more explicitly, using a well-attested stage of truncation, from English among other languages.

This example comes from the development of word shape and syllable truncation, which has been central to previous work on stages of acquisition. The relevant stage is one where children have abandoned truncation to one *syllable* (stage 1 below), and now allow outputs up to but no bigger than one *foot*.

101) *the initial stage*: /σ/, /σσ/, /σσσ/ → [σ]

the intermediate stage: /σ/ → [σ]; /σσ/, /σσσ/ → [σσ]

This intermediate stage has been attested among children learning a variety of languages: Dutch (Fikkert, 1994; Demuth, 1995; see also Lohuis-Weber and Zonneveld, 1996); English (Pater, 1997; Gerken, 1994; Saladis and Johnson, 1997; Kehoe, 2000); German (Lléo and Demuth, 1999); as well as Greek (Revithiadou and Tzakosta, 2004); Spanish (Lléo, 1996; Gennari and Demuth, 1997). Some representative examples are given below, from German and English:

102) *The intermediate stage: one foot outputs*

	Target	*Child*	
English (Kehoe, 2000, table 3)	'bunny' 'giraffe' 'banana' 'elephant'	[bʌ́ni] [dɛf], [ʔæf] [nǽnæ] [áfɪnt], [ɛ́lbɪnt]	subject 18m4 (1;6)
German (Lléo & Demuth, 1999)	'kaputt' 'Karton' 'Geburtstag' 'Kartoffel'	[púχ] [tón] [búdzɑ], [búdas] [tófɛl]	Marion (1;10.5) Thomas (1;9.0) Marion (1;11.25) Johannes (1;9.21)

Before discussing the analysis, a crucial data caveat. Shelley Velleman (p.c.) points out that the literature outside Germanic does not provide much support of this initial stage of truncation to monosyllables. In fact, see e.g. the results for Finnish, French and Italian in Vihman (2001) which suggest that for learners of these languages, the initial stage of production already includes two-syllable outputs (and in fact treats disyllables as the minimal word, i.e. mapping /σ/ → [σ σ].) Thus, the monosyllabic state being discussed here is not being claimed as the truly initial state, but rather reflects an early state which learners of English, Dutch etc. are commonly in at the onset of word production. (Recall from the discussion in 4.2.2 above that some grammatical learning is assumed to have taken place before production begins.)

The analysis that I propose for these two stages does not rely on a stringency relation between two markedness constraints. Rather, I suggest merely that the one-syllable stage reflects one aspect of optimal satisfaction of the conflicting constraints on foot form and alignment that begin at the top of the learner's M >> F ranking, and that stage 2 reflects the error-driven demotion of a foot form constraint. And as we will see, the Error-Selective Learner can easily pass through this stage – not because the relevant

constraints are in a stringency relationship, but because of the frequency of violation of wordshapes.

At the initial stage, these learners reduces outputs to a single syllable. If we interpret this syllable as a bimoraic foot, this output satisfies a number of prosodic constraints: the need to align all feet with the word edge, and the demands of *both* foot form constraints, Trochee and Iamb (thanks to John McCarthy for suggesting this analysis):

103) *Prosodic Markedness constraints*

a) All-Ft-L: "The left edge of every foot is aligned with the left edge of a Prosodic Word"

b) Trochee: "Heads of feet must be left-aligned in the foot"

c) Iamb: "Heads of feet must be right-aligned in the foot"[53]

In the ranking below, Max ranks below all of these markedness pressures, so the winning candidate in the tableau of 103) is the single syllable output in (i). For illustration's sake, I illustrate this with a three syllable word with medial stress, although the ranking generalizes to other multi-syllabic inputs:

104) Stage One: All-Ft-L, Trochee, Iamb >> Max

[53] There are clearly other differences between trochees and iambs than the alignment of their heads – see e.g. Hayes (1995). I assume these differences are the result of other constraints.

105) *Stage one: one syllable only*

/σσσ/	All-Ft-L	Trochee	Iamb	Max[54]
(i) ☞ (**σ**$_{\mu\mu}$)				**
(ii) (**σ** σ)		*!		*
(iii) (σ **σ**)			*!	*
(iv) σ (**σ** σ)	*!		*!	

106) *The resulting ERC row*

	All-Ft-L	Trochee	Iamb	Max
σ(**σσ**) ~ (**σ**$_{\mu\mu}$)	L	e	L	W

At the intermediate stage, feet must still be aligned to both word edges (so there can still only be one of them). What characterizes the move to this stage, however, is that the language-specific foot type – either trochees or iambs – has been acquired, via demotion of the conflicting markedness constraint. In English, this intermediate stage will now treat the word-medial case above by producing a bi-syllabic trochee (tableau 108 below) – but it will still reduce any word with two feet to only one (109):

107) Stage Two: All-Ft-L, Trochee >> Max >> Iamb

108) *The intermediate stage: one trochaic foot…* [55]

/σσσ/	All-Ft-L	Trochee	Max	Iamb
(i) (**σ**$_{\mu\mu}$)			**!	
(ii) ☞ (**σ** σ)			*	*
(iii) (σ **σ**)		*!	*	
(v) σ (**σ** σ)	*!	(*)		(*)

[54] Note that I am calculating Max violations here in terms of syllables only because the candidates have been simplified to syllables. The real Max constraint I am assuming in fact counts violations by segment, but nothing crucial hinges on that here.

[55] The fact that outputs do not contain any post-tonic syllables at this stage, i.e. [(σ)σ], can be attributed to Parse-σ (Prince and Smolensky, 1993) or Lapse constraints (Elenbaas and Kager, 1999**)** constraints, that ultimately prefer to delete syllables if they cannot be footed.

109) *... and one foot only*

/ˌσσˈσσ/	All-Ft-L	Trochee	Max	Iamb
(i) (σμμ)			**!	
(ii) ☞ (σ σ)			*	*
(iii) (σ σ)		*!	*	
(iv) (σσ)(σσ)	*!			*

The question is why children should decide to demote Iamb before All-Ft-L, and the Error-Selective answer comes from the frequencies of the forms that cause these two constraints to assigns Ls. To see this, we must consider the kinds of errors in which these constraints have different violation profiles.

Since English is a fully trochaic language, Iamb is rampantly violated in the words that children hear. One very common English word shape is the trochee itself – bisyllabic, with initial stress. And these words will create ERC rows in which Iamb assigns an L but All-Ft-L does not, since both the winner and loser's only foot is indeed left-aligned:

110) *The kind of English ERC row in which only Iamb assigns an L*

'máma'	All-Ft-L	Trochee	Iamb	Max
(σσ) ~ (σμμ)	e	e	L	W

However, English is also a language with iterative footing, so that many winners violate All-Ft-L as well. But as we've seen above, the only English foot type that does not violate this definition of Iamb is a heavy monosyllable, so it will only be words with *two* monosyllabic feet to which only All-Ft-L will assign an L:

111) *The kind of English ERC in which only All-Ft-L assigns an L*

'pòntóon'	All-Ft-L	Trochee	Iamb	Max
(ˌσ)(ˈσ) ~ (σμμ)	L	e	e	W

Since the Error Selective learner demotes constraints only once they reach the Violation Treshold, the question which type of ERC row will be more frequently represented in the Support? It is hopefully not contentious that children encounter English words like "mama" more frequently than words like "pontoon". As we already saw in the Roark and Demuth (2000) findings of §4.2, 90% of their corpus of English child-directed speech contained tokens that were either monosyllabic (violating neither constraint) or disyllabic trochees (like the ERC row in 110). From this alone we should expect that Iamb will reach its Violation Threshold before All-Ft-L, and so add an error like 110) to the Support to create the attested intermediate stage.

To recall the more general point, then: stringency between markedness constraints is not in any way necessary for the ESL learner to pass through an intermediate stage of constraint ranking.

4.3.1 Noting an stringent alternative

It should be noted that Curtin and Zuraw (2001) in fact derive the one-foot intermediate stage using two markedness constraints on prosodic structure that sit in a stringency relation:

112) *The Specific M analysis of the one-foot stage* (from Curtin and Zuraw, 2001)
All-Ft-L >> Max >> All-σ-L

However, the less stringent constraint that they use is the somewhat implausible 'All-Syllable-Left', which requires every syllable to be aligned to the left edge of the Prosodic Word. The cross-linguistic support for this constraint is not particularly robust; those patterns that produce maximally one-syllable outputs may well be dealt with using the kind of prosodic constraints assumed in the analysis above (on prosodic maximality, see McCarthy and Prince, 1993; Ito, Kitigawa and Mester, 1996; Ussishkin, 2000; de Lacy, 2004.) I will return to Curtin and Zuraw's analysis in chapter 3 §3, however, in the discussion of how stringency relations among faithfulness constraints shape the stages of GLA learning.

4.4 Infrequent mistakes and the value of the Error Cache

In the original BCD model, a one-time mistake in the data can in fact threaten the entire delicate search for restrictiveness. For example, if the BCD learner of a language with mid vowels only in stressed syllables happens to hear e.g. a slip of tongue with an unstressed mid vowel and add it to their Support, it will end up with the over-generating grammar that we have been trying so scrupulously to avoid.[56]

We have already seen that adding a Cache to the BCD learning procedure means that not every error the learner makes will be learned from; in fact, many errors will be added to the then-current Error Cache, but never get transferred to the Support. So in the Error Cache, the learner also has a place to keep temporary track of the frequency of individual ERC rows – that is, how many times a particular winner-loser pair has been seen. Thus, one could include an initial criterion in the ESA saying that a best error (i.e. a

[56] As pointed out by Boersma and Hayes (2001), this ability to be 'robust' in the face of noisy data is a virtue of the Gradual Learning Algorithm – see chapter 3.

best error *type)* can only be one that has been made more than some minimum number of times (i.e. *tokens*.)

If we make this move, then to be make sure that the learner will still end up eventually with an empty Cache – see §3.2.4 above – we must also add some requirement ensuring that any ERC row which has been heard fewer than a certain number of times over a certain amount of time is erased from the Cache without any other impetus. In other words, hearing an insufficient number of tokens of a particular error type will lead the learner to decide that ERC row was just noise. Error-Selective learning makes this approach possible, unlike in BCD, because it decouples the reason for re-ranking, which is the current grammar's errors, from the trigger of re-ranking, which is exceeding the Violation Threshold for some constraint. This gives the learner some leeway to ignore infrequently-made errors.

A very useful effect of keeping track of token as well as type frequencies in this way is that the Error-Selective learner can make the crucial distinction between (i) noisy data, which should never be transferred from Cache to Support, and (ii) grammatical exceptions, which should. Suppose that the learner is acquiring a language where a very few lexical items have codas – perhaps only three very recent borrowings – but 99.9% of the lexicon is coda-free. To be properly robust, the learning algorithm must be able to distinguish the Support for this exceptional coda language from one in which codas are 100% ruled out, even if the learner has misheard three words in the latter language as having codas. The difference will be found in their token frequencies. In the former language, only three ERCs can demonstrate the exceptionality of NoCoda but this exceptionality will be demonstrated *every* time each of these lexical items is heard,

whereas in the latter language the misheard codas will be one-time events. Thus as the former learner's VT sinks towards one, the three errors demonstrating the exceptional need for NoCoda >> Max will eventually trigger learning and get added to the Support, prompting some change to the grammar (recall chapter 1 §2.2.) In the latter case, however, by the time the VT gets low enough the Error Cache will already have been emptied of the misheard 'codas', just for having been heard only once.

5. Developmental variation and Error-Selective Learning

Perhaps the largest idealization made in the learning discussion of this chapter has been the abstraction away from any output variability in child data. The empirical reality is that children's outputs are in fact variable in a number of ways: that at any one stage of acquisition, children produce the same words or phonological structures in a variety of different ways.

As one example to use in the discussion that follows, I return to the first intermediate stage discussed in chapter 2 in which singleton codas have been acquired but complex ones have not. One of the children in table 3) of §2.2.1, P.J., was in fact at a stage where input singleton codas were only sometimes preserved, and other times deleted. Looking back at the data from Trevor and Julia's syllable margins month by month in section 3, it is clear that both children went through many months of variable singleton coda deletion. And after mastering faithfulness to singletons preservation, they later also passed through a stage of variability in their production of *complex* codas.

5.1 The ubiquity and challenges of variation in learning

The issue of where or how developmental variation should be captured by a grammatical learning theory does not seem straightforward. Broadly speaking, I see two ways into the problem. One is to attribute variability in development to the learning mechanism, and not to the grammars constructed by those mechanisms. The other way is to make variation an inherent property of the grammars per se: as we will see in detail in chapter 4, this is the nature of the stochastic OT approach and the associated Gradual Learning Algorithm (Boersma, 1997).

A third position worth considering is the possibility that all variation in learning is the result of performance problems. Under this view, learners whose grammar has just re-ranked so as to permit coda consonants must still learn to produce the necessary articulatory gestures associated with those codas. It seems reasonable that articulatory pressures are responsible for some of the variation that learners display, and I do not have any perfect arguments as to why they could not explain *all* variation. I note, however, that the connection between input frequencies and order of acquisition does not appear to hold in the case of marked structures that present clear articulatory problems. For example, the English interdental fricatives are notoriously difficult to produce, and while they are extremely frequent in English inputs they are quite late to be acquired. Thus, it might be possible to diagnose a kind of variation that is attributable to performance problems, and still find other evidence of grammatical variability left unaccounted for; I leave this tentative suggestion as a question for further research.

In some sense, the most extreme version of the performance problems view is to abandon the notion of children's outputs as involving phonology at all. The claim is that

the amount of attested variation indicates that constraint rankings are not responsible for any stages of production; this is at least the position of Hale and Reiss (1998). While this tack leaves us fewer things to explain, it does so at the expense of understanding several things. First, it does not give us any way of explaining the ways in which children's outputs are not *more* variable – that is, that they are stable and systematic, at all but perhaps the earliest stages (c.f. Ferguson and Farwell, 1975). A related, more specified analytic disappointment is that it writes off the observation that children's developing grammars can often mirror and innovate patterns found in the typology of natural languages – including those beyond the target – as an accident of flapping meat and phlegm. And third, a performance-only view can not explain why children's innovative patterns and errors can reflect sensitivities to abstract properties such as the notion of morphological basehood (see evidence of such innovations and discussion of this point in chapter 4 §7.2.)[57]

5.1.1 The potential for a variable BCD learner

How can developmental variation be treated in the present system? As I have already stressed (or perhaps conceded) this dissertation is no way an empirical study of variability in phonological learning. But since Error-Selective learning is an attempt to model more of the human acquisition process than pure BCD, we should at least ask to what extent variability across stages can be captured by this theory.

Given that the grammars my BCD algorithm learns do not contain any variation, my error-selective learner can only demonstrate variation through some elaboration of the learning procedure. This BCD algorithm builds what I will refer to as ordinal rankings –

[57] Thanks to Joe Pater for pointing out this argument to me.

that is, each constraint ranks above or below another -- e.g. C1 >> C2 -- but there is no sense in which C1 can be *more or less* ranked above C2. As I will discuss in some detail in chapter 3, other theories of learning assume an OT grammar in which constraints are ranked on a numerical scale, so that it IS possible for C1 to be ranked a lot or a little above C2 – this is true of the Gradual Learning Algorithm (Boersma, 1997 *et seq.*) We will see in chapter 3 that the possibility of one constraint outranking another one just a little bit is how the GLA learner naturally shows variation between its intermediate stages over the course of learning.

If we are committed to an ordinal OT grammar – which all extant versions of BCD such as the one I have adopted here certainly are – then our learner does not have any way to build *rankings* that encode any degrees of vacillation between intermediate stages, analogous to the GLA. Instead, however, we can consider how we could modifiy the error-selective BCD learner's methods in order to derive the effects of variation between rankings. In the rest of this section I will suggest two such possible methods: neither is presented as a definitive approach to variable Error-Selective Learning, but together they may provide future areas of investigation for the model.

The first alternative is to change the notion of a Violation Threshold from a fixed value to a range of values – this means that it will sometimes be easier to trigger learning and add new errors from the Cache into the Support than other times. If the learner temporarily adopted a low VT, they would add more errors to their Support and so build rankings that appear to represent a later stage of development. If at the same time the learner also remembered that the VT that allowed those errors into the Cache was lower than normal, they could periodically empty their Support of such suspect errors, and thus

build a ranking that reverts to an earlier stage. An initial implementation of this approach is given in 5.2 below; in section 5.2.4 I raise a few ways in which a more realistic version of this variable learner could be built.

A different idea about variation in ESL would be to suggest that the Support is not the single repository of permanent errors that I have been claiming it to be thus far. Instead, this variable learner would learn from a best error not by adding that error into the one Support but *cloning* the previous Support and adding the new error to that clone. In this approach, every cycle of learning would build a new Support (based on the previous one), BCD would be used to build a ranking tied to each Support clone, and learners could pick (randomly or otherwise) from their current ranking options in order to process new data. Over time, each Support would decay in memory as a function of how many errors it still made: the more errors, the quicker the memory loss. Once a Support was forgotten, its ranking would be forgotten, too, and so over time the older rankings would disappear from use and the newer ones would gain credence. This idea is briefly explored in section 5.3.

5.2 Alternative I: the Variable VT approach

As discussed in section 4, the introduction of an Error Cache has consequences for any aspect of learning that is in some way temporary. What I will explore here is the notion that errors in the Cache could derive the effects of later stage rankings by being temporarily introduced into the Support, but not retained because they have yet to truly overcome the Violation Threshold.

5.2.1 The example of variable codas

Recall the first Error Cache used to illustrate ESL of codas vs. complex codas, repeated below in 113). Up until now we have treated the Error Cache as inert up until the point when some constraint exceeds its Violation Threshold. So if, for example, our Violation Threshold is set to 4, then the Error Cache below is *about* to trigger learning on NoCoda but hasn't yet, and none of its errors have yet had any effect on re-ranking:

113) (repeated from 69)

Input	*Winner ~Loser*	***NoCoda***	**CompCoda*	*Max*	**CompOnset*
i) /frend/	frend ~ fɛ	**L**	L	W	L
ii) /piz/	piz ~ pi	**L**	e	W	e
iii) /greɪp/	greɪp ~ ge	**L**	L	W	L
iv) /ti/	ti ~ si	e	e	e	e

In the variable ESL approach, however, overcoming the true Violation Threshold is not actually necessary to trigger the inclusion of an error into the Support. Imagine instead that every time the learner uses the grammar they adopt a *temporary* Violation Threshold, that may be different than the true VTs (more on how this works in a minute.) If a temporary VT is lower than the true one, it may already be met or exceeded by some constraint in the Cache and thereby trigger an early application of the ESA. This early version of the ESA analyzes the Cache to find a best error – one which violates a Triggering Constraint according to the temporary VT. The learner will then *copy* this error (rather than *move* it as in normal ESL) to the Support.

In this system, a temporary trigger constraint is one whose number of Ls meets or exceeds the *temporary* VT; thus in 113) above, NoCoda is a temporary trigger constraint,

because its three Ls in the Cache meets the temporary VT. With this slight re-definition of triggering, the Early ESA is otherwise exactly the same as the original in §3.1:

114) *The Early ESL Algorithm*
Choose as the best error that row in the Cache which:
a) has an L assigned by the *Temporary* Trigger Constraint
and of those, the one that
b) has the fewest Ls assigned by other Markedness constraints
and of those, the one that
c) has the most Ws assigned by Faithfulness constraints

As we saw when we were assuming a VT of 3: the Best Error in the Error Cache above is candidate (ii) "peas", because it violates the temporary Trigger Constraint (NoCoda) but no other Markedness constraints. And in this Variable ESL scenario, "peas" is now the temporary Best Error.

115) *Using temporary violation thresholds to trigger Early ESA*
The true Violation Threshold: 4
The temporary Violation Threshold: 3

Temporary V.T triggers learning

b) NoCoda is the Temp. Trigger Constraint

Input	*Winner ~ Loser*	***No Coda***	******Comp Coda*	*Max*
/frɛnd/	frend ~ fɛ	**L**	L	W
/piz/	piz ~ pi	**L**	E	W
/greɪp/	greɪp ~ ge	**L**	L	W
/ti/	ti ~ si	E	E	e

early ESA

d) the Error Cache NOT cleared…

Input	*Winner ~ Loser*	*No Coda*	******Comp Coda*	*Max*
/frɛnd/	frend ~ fɛ	**L**	L	W
/piz/	piz ~ pi	**L**	e	W
/greɪp/	greɪp ~ ge	**L**	L	W
/ti/	ti ~ si	e	e	e

c) the pre-existing Learning Support Table

Input	Winner ~ Loser	No Coda	*Comp Coda	Max
... empty, waiting...				

e) … but the Support IS updated:

Input	Winner ~ Loser	No Coda	*Comp Coda	Max
/piz/	piz ~ pi	**L**	e	W

Although this error's Trigger Constraint may not yet have exceeded the true VT, and although the Cache has not yet been cleared – adding this Best Error to the Support will still trigger step 2, and so BCD will build a new ranking:

116) *The BCD ranking resulting from the new Support in 115)e):*
*ComplexCoda >> Max >> NoCoda

This new ranking in 116) has brought the learner to the intermediate stage of coda acquisition – but it is not a *stable* grammar because the learner has not yet seen enough errors to overcome the true VT and permanently demote NoCoda. In the Variable ESL system this instability has been encoded by copying rather moving temporary errors, leaving them in both the uncleared Cache AND the Support: while the learner relied on a temporary error to get their current ranking, they are not committed to its permanence in the Support. In the Variable ESL system, this instability is resolved at the end of each use of the current ranking using a process of 'Synching' the Error Cache and the Support. To perform this Synch, the learner compares the Error Cache and Support, finds any identical errors that appear in both, and *removes them from the Support.* This has the effect of forgetting any temporary errors that have been included in the Support due to a temporary Triggering Constraint.

In the example above: after building the ranking in 116), the learner synchs the Cache and Support, finds the identical ERC "peas" in both, and so deletes it from the Support:

117) *Synching the Cache and Support*

d) the uncleared Cache:

Input	*Winner ~ Loser*	*No Coda*	**Comp Coda*	*Max*
/frend/	frend ~ fɛ	**L**	L	W
/piz/	piz ~ pi	**L**	e	W
/greɪp/	greɪp ~ ge	**L**	L	W
/ti/	ti ~ si	e	e	e

e) the Support, with an identical error:

Input	*Winner ~ Loser*	*No Coda*	**Comp Coda*	*Max*
~~/piz/~~	~~piz ~ pi~~	~~**L**~~	~~e~~	~~W~~

Now the Cache and the Support have been synched, the Support's evidence for Max >> NoCoda has been *removed,* so while the current ranking is at the intermediate stage of coda acquisition, the Support has returned to the previous stage. This means that the *next time* constraints get re-ranked (more on that below), the learner's ranking will follow the Support in reverting to the previous stage.

Where do temporary Violation Thresholds come from? Inspired by the stochastic OT system used by the GLA, where each run of EVAL randomly draws a value for each constraint from its probability distribution, I suggest that the Variable ESL begins each use of the grammar by similarly choosing a temporary VT value from a (normal) probability distribution, whose mean is the true Violation Threshold.[58]

[58] If it turned out that different constraints should be assigned different Violation Thresholds, we could center this normal distribution around the *mean* of all the true VTs. For example, we might set the VTs for prosodic constraints lower than segmental ones, to derive the fact that learners acquire their stress systems much earlier than their full segmental inventories.

5.2.2 The effects of the Variable VT approach

I began this section with the idea that the Variable ESL learner chooses a temporary Violation Threshold every time the grammar is *used,* not only when it makes an error. So let us now step back and see how this system works as a whole, and then consider its pros and cons.

Every time the learner has used the grammar and chosen an optimal output for some input, they determine whether their output was an error or not. If they haven't made an error, they do nothing. If they have, they add the error to the Cache, generate a temporary VT, and then check whether any constraint in the Cache is now a trigger constraint. If not (that is, if no constraint has assigned as many or more Ls as the temporary VT value), then the learner simply goes to the existing, unrevised Support and builds a new ranking via BCD to be used next time. (This re-ranking gets the grammar back in synch with the Support, in case the last re-ranking was done using a previous temporary error.)

If the VT value chosen *has* been exceeded by some constraint or set of constraints, then the learner must add a best error to the Support. To know how to do so, the learner checks whether the temporary VT value is equal or greater than the true VT.[59] If it is, then the learner behaves as he or she would have in section 3: uses the normal ESA algorithm, *moves* the chosen best error into the Support, clears the Cache, and builds a new ranking via BCD to be used next time. If however the temporary VT value is *less* than the true VT, the learner instead uses the variable ESA algorithm, *copies* the best error into the Support, does NOT clear the Cache and again builds a new ranking via

[59] … or the true VT for the relevant constraint, if assuming different ones.

BCD. Finally, the learner synchs the Cache and the Support so that a copied best error will be removed again from the Support.

The first thing to say about this proposal is that it has at least done what we set out to do. That is: adding a variable notion of the Violation Threshold, and synching the Cache and Support after learning, will indeed get us the effects of variation between stages. At the end of each learning cycle, the Support of a Variable ESL learner is in the same state that it would be under normal ESL. What's crucial is that its ranking may be different: if a low temporary VT led the learner to choose a temporary best error, the new ranking will reflect that error's ranking entailments but the Support will already have forgotten them (via synching). In the above example: the learner has built a grammar where singleton codas are preserved faithfully, but it has forgotten the error that enforced this faithfulness. For the moment the learner appears to have acquired singleton codas, and his grammar will parse them faithfully. But the next time the current grammar makes *any other* error a new ranking will be built from the Support, and the learner will return to a state of coda deletion. Through this flip-flopping of contents in the Support, this system derives variability between different rankings.

This extension of this model has also not sacrificed anything integral to the original ESL proposal. As in the original ESL proposal: when the *true* Violation Threshold is exceeded by some constraint, errors will permanently be moved into the Support, and all future rankings will reflect that move. Because the learner clears the Cache after applying the normal ESA, synching will not find any identical errors in the Cache and Support, so nothing will be deleted. And given that BCD will always choose a

ranking that prefers winners over losers, errors already in the Support will not be made again, so the Cache won't get re-cluttered with errors it has already (truly) learned from.

5.2.3 Deriving developmental regression in the variable VT approach

It might also be the case that the variable VT approach could be extended to explain apparent cases of regression, if we allowed the learner to adopt a low temporary VT *for an extended period of time* rather than choosing one after each new error. For example, if the learner above repeatedly chose a temporary VT of 3 for NoCoda they could appear to have fully progressed to the singleton coda stage – NoCoda would be continually triggering learning and adding errors with codas to the Support, and the synching progress would be removing those errors from the Support again, just as continually. If then the learner abandoned their temporary threshold and re-adopted the true, higher VT, NoCoda would stop triggering learning until the real VT was met, and in the meantime our child would appear to have regressed back to the coda-less initial stage.

What exactly would prompt the learner to adopt this lower VT for a long period of time, and why they would later revert to the true VT, remains unclear. As suggested in section 3.2.4 above, this proposal would be supported by evidence that children's regressions coincide with increased demands on their cognitive resources more generally – for example with the advent of a burst in lexical acquisition.[60]

[60] See the somewhat related arguments in Stager and Werker (1997) and Fennell and Werker (2003) about the connection between decreased phonemic discrimination in tasks that pair sound and meaning among infants who have reached a stage of increased lexical acquisition (around 14 months). See also Pater, Stager and Werker (2004) for discussion of OT implementation of the relationship between cognitive load and variable rankings.

5.2.4 Weaknesses of the variable VT appraoch

This section has sketched one direction in which a variable ESL learner might evolve, but this approach does not satisfactorily address all the issues. For one thing, the use of variable VTs is somewhat stipulative: especially because if the learner can always reference what the real VT is and use it to re-synch the Cache and Support, it is somewhat unclear why they would periodically choose a temporary lower one. Furthermore, although picking the temporary VT from a normal distribution predicts that most temporary values chosen will be clustered around the true VT value, this approach doesn't really connect the degree of evidence the learner has for ranking with the likelyhood that they use that ranking at any given time.

5.3 Alternative II: the Cloned Support approach[61]

In addition to the variable VT idea, section 5.1.1 also raised a second possibility about developmental variation in BCD learning. This alternative retains the original ESL ideas of a single, true Violation Threshold to trigger learning, and a single mechanism by which chosen errors are added permanently to a Support and the Cache cleared. What is different in this account is the conception of the Support itself. This learner uses each new best error to to build an alternative Support, which contains all the old errors plus the new one, and which is kept in memory alongside the previous one. Each Support is used by BCD to build a grammar, and each time the learner goes to produce a new output they can choose any of the currently-held grammars to feed it through. Thus in this model, the learner varies between stages because they vary their choice of stored grammar to use.

[61] Thanks to Lyn Frazier and Jonah Katz for comments that inspired this approach.

5.3.1 Returning to the variable coda example

To see how the cloned Support approach works: suppose that our learner's Violation Threshold is 4, and that the learner has just added an error to their Cache that will trigger learning on NoCoda

118) *An Error Cache in which NoCoda overcomes the VT:*

Input	*Winner ~Loser*	***NoCoda***	**CompCoda*	*Max*	**CompOnset*
i) /frɛnd/	frɛnd ~ fɛː	**L**	L	W	L
ii) /piz/	piz ~ pi	**L**	e	W	e
iii) /greˡp/	greˡp ~ ge	**L**	L	W	L
iv) /tost/	tost ~ to	L	L	W	e

Looking at these errors, we can see that while the learner has yet to learn much of anything about English syllable structure, it has already acquired some simple facts about the English segmental inventory – for example, that mid vowels and labial consonants are all allowed. This means that some errors demonstrating a tolerance for these marked features must have already made it into the Support (as in 119a below), building a grammar like in 119b):

119)a) *An existing Support for the learner in 118)*

Winner ~ Loser	*Mid	*Lab	Ident [mid]	Ident [lab]	No Coda	*Comp Coda	Max
béʲbi ~ dídi	L	L	W	W	e	e	e

119)b) *A grammar that BCD builds from 119a)*[62]

NoCoda, CompCoda >> Id[lab] >> Id[dors] >> *Mid, *Lab >> *Max

As discussed several times already: the ESL learner faced with the Cache in 119) will choose 'piz ~ pi' as the best error to learn from, and up until now that has meant updating the Support with this error. Instead, this alternative learner uses the best error from 118) to build a clone of the Support in 119)a), and build another ranking from that clone. This means that after NoCoda overcomes the violation threshold in 118) and a cycle of learning has occurred, the learner has TWO Supports, as in 120) below, and thus that its grammar contains TWO different rankings as in 121):

120) *The state of the cloned Support learner after NoCoda triggers learning in 118)*

a) *Support A – pre-existing*

Winner ~ Loser	*Mid	*Lab	Ident [mid]	Ident [lab]	No Coda	*Comp Coda	Max
bé^{i}bi ~ dídi	L	L	W	W	e	e	e

b) *Support B – cloned Support A plus one new error*

Winner ~ Loser	*Mid	*Lab	Ident [mid]	Ident [lab]	No Coda	*Comp Coda	Max
bé^{i}bi ~ dídi	L	L	W	W	e	e	e
piz ~ pi	e	e	e	e	L	e	W

[62] A reminder of how BCD gets a grammar like this from 117a). First we install all M constraints with no Ls (those against syllable structure); then we have to install one F constraint that assigns a W, so we install Id[dors] to free up *Dors in the next stratum. Then we again have to install F constraints until an M constraint is available, which means Id[mid],[lab] to free up *Mid, *Lab, and then we install the remaining F constraint Max and we have a grammar.

121) *The resulting grammar with two rankings*

a) from Support A: **NoCoda,** CompCoda >> Id[mid] >> Id[lab] >> *Mid, *Lab >> **Max**

b) from Support B: CompCoda >> **Max >> NoCoda** >> Id[dors] >> Id[lab]>> *Mid *Lab

The bold face and underlined constraints are those in conflict with each other. This is to make clear that the first ranking in 121a) is one where segmental restrictions have been overcome (F >> M) but syllable structure remains fully unmarked (M >> F) – while in 121b) some syllable markedness has also been acquired (specifically Max >> NoCoda).

In this ESL model, the learner now has two rankings as part of their grammar, and every time it uses its grammar it must first pick one of its rankings. When it picks the one built from Support B, it produces singleton codas faithfully; when it picks the one built from Support A, it still deletes all codas. And thus it vacillates between two intermediate stages.

Note that to remain in line with the goals of this dissertation, our learner must still be remembering Support(s) as its primary data rather than rankings – so, we can say that though the learner has multiple Supports in memory simultaneously, it also knows which ranking comes from which Support, and as soon as any Support is forgotten its associated ranking disappears as well.

Thus, the necessary second part of this cloned Support model is how the learner gets rid of old Supports. The basic proposal is that each Support decays in memory in proportion to how many errors it prompts the learner to make. One way to implement this idea would be that the learner keeps a "reliability score" associated with each current

Support hypothesis[63] – suppose we start each freshly-cloned Support's score at 1 (meaning 100% reliability). The first time a particular Support's ranking is used to process an observed form and makes an error to add to the Cache, that Support's reliability score is lowered: perhaps by a fixed amount (say to 0.9), or perhaps more intelligently as a proportion of the number of errors in that Support.

In this second scenario: suppose the freshly-cloned Support B in 120) had a reliability score of 1 and we then used Support B's ranking to make a new error like [tost] ~ [tos]. The learner would now have the Support's two resolved errors in favour of the ranking (on 'baby'), and one new error against it (on 'toast'), so its reliability score would now be 0.5 (1 out of 2).

Finally: once a Support's reliability score sinks low enough it is forgotten altogether, and its associated ranking disappears as well. In the case of 120), the ranking built from Support A makes all the same errors as that built from Support B – *plus* errors on singleton codas. Thus Support A's reliability score will sink faster than Support B's. Once Support A is forgotten, the learner will have moved out of the vacillation stage, and always produces singleton codas faithfully from now on.

5.3.2 Discussion of the Cloned Support approach

This alternative provides a different view of variation in ESL than the variable VT approach. This most recent learner does not vary between stages because the contents of their single Support grows and shrinks again, but their *set* of Supports grows and shrinks. One benefit of the cloned Support approach is that it does not require any selective amnesia of the true VT; nor does it require any process like synching.

[63] The idea of a reliability score comes quite directly from Albright and Hayes (2003)'s rule-based learner.

In this Support-cloning view, all variation is dictated by the order and speed with which new Supports are created and old Supports are forgotten. New Supports are still built in the normal ESL fashion: i.e. when learning is triggered by some constraint overcoming the VT in the Cache. Meanwhile, old Supports are forgotten via their reliability score. This ensures that older Supports – ones that have fewer target rankings and so prompt more errors – are forgotten quickly and newer Supports are retained longer. In the end, the learner's final Support will retain a perfect reliability score – because it never makes any new errors.

5.3.3 Regression in the Cloned Support approach

Another use of the reliability score could be to influence the learner's choice between its multiple current Supports in processing new data – the higher a reliability score, the more likely the learner could be to use that Support's ranking. A side benefit of connecting a Support's reliability with its ranking's continued use might be that quirks in the data could create regressions to earlier stages.

To get regression in the cloned Supports model, the learner would have to get hung up on using an older Support rather than a newer one for a period of time. This would mean that an older Support would need a higher reliability score than its competitors, which could happen temporarily as a fluke of randomization. Suppose that two new Supports have just been created, so that each has a near-perfect reliability score and each associated ranking is being chosen about as often as the other. If it happened that most of the observed forms fed to the slightly older Support were relatively unmarked, while most of the marked forms were fed to the slightly newer Support, the

misleading upshot would be that the newer Support was *less* reliable. And for a short while – until the errors of the older Support caught up – the learner could appear to have regressed to an earlier stage.

5.4 Summarizing the variable ESL discussion

This section has presented some issues and ideas for extending the Error-Selective learner to model developmental variation between stages. I suggested two different ways in which the general proposal could be modified, by either adding some errors to the Support in a temporary way (§5.2), or by building multiple, temporary Supports (§5.3)

One point about both suggestions is that these variable ESL learners clearly treat variability between stages of acquisition differently than variation at the end-state grammar. Once errors are no longer being made, there will be no more errors in the Cache to violate Violation Thresholds or trigger Support clonings – so there will be no vacillation between rankings. This contrasts sharply with the GLA approach to be discussed in the next chapter, in which variation between rankings is an inherent property of every grammar: developing, stable or otherwise. The extent to which the variability seen in developing vs. adult grammars should be treated as a unified phenomenon is not necessarily clear – in part because adult speakers can overtly control their choice of variants with respect to socio-linguistic factors, in a way that a child varying between the codaless and singleton coda grammars clearly does not. Still the BCD learner's treatment of any kind of variation remain tenuous enough to require further work; after my discussion of the GLA, I return to the issues of end-state variation and BCD-style learning in chapter 3 §5.5.

6. Chapter Summary

The goal of this chapter has been to introduce Error-Selective Learning, as a framework for gradual learning using BCD. I have discussed at length the ways in which ESL uses properties of ERC rows and their frequency to slowly add errors to the Support, which in turn slowly provides evidence to the learner of the target grammar. I have exemplified the approach and the stages it provides using a number of constraints and languages from the literature, which I hope will have demonstrated its breadth. I have also introduced two alternative ideas for how the Error-Selective Learner could vary between stages in a gradual way, and even show the temporary effects of developmental regression. The best way to incorporate variation into the ESL model, particularly with the BCD's view of constraint rankings, still remains to be seen; see also chapter 3 §6.