Stages of Phonological Acquisition and Error-Selective Learning

Anne-Michelle Tessier
University of Massachusetts Amherst

1. Introduction

This paper presents an error-driven model of Optimality-Theoretic acquisition, called Error-Selective Learning, which is both restrictive and gradual. It is restrictive in that it chooses grammars that can generate observed outputs but as few others as possible, using a version of Biased Constraint Demotion (BCD: Prince and Tesar, 2004). It is gradual, unlike a pure BCD learner, in that it requires numerous errors of the same kind to learn a new grammar. Together, these two properties provide a model that can derive many observed intermediate stages in phonological development, while still explaining how learners eventually converge on the target grammar.

In this paper, I present evidence of two intermediate stages that are frequently observed in L1 acquisition (§2), and demonstrate how both can be captured with rankings that sit between the initial and final stages with respect to the position of their Markedness constraints. Section 3 introduces the notion of learning grammars from errors via the BCD algorithm, and shows how BCD learns too fast and efficiently to derive intermediate stages. I then introduce Error-Selective learning (§4), showing how this learner chooses the errors it learns from, and demonstrating how its choice of errors derives the attested intermediate stages. Section 5 provides some brief discussion, and section 6 concludes.

2. Representative data from two intermediate stages

A well-established fact about phonological acquisition is that children's initial grammars are unmarked, broadly speaking, and that learners acquire the marked aspects of their language over time (e.g. Jakobson, 1941/1967; Stampe, 1969; Smith, 1973; Fikkert, 1994.) In Optimality Theory, the resulting proposal about the initial state of learning is that markedness constraints begin ranked above faithfulness constraints: M >> F (e.g. Gnanadesikan, 1995; Demuth, 1995; Pater, 1997, see also Smolensky, 1996.)

The literature also provides much evidence that between initial and final grammars, children pass through various intermediate grammars that capture some but not all of the target grammars's markedness. In this section I demonstrate data from two such data patterns, and the rankings that describe them. These are the stages that the Error-Selective Learning proposal of section 3 will derive.

2.1 The Specific-M stage and the acquisition of Germanic complex codas

Children acquiring languages with complex coda structures, such as English and Dutch, often go through an intermediate stage where singleton codas are preserved faithfully, while coda clusters are reduced to singletons. This stage has been documented in Dutch for e.g. Eva at 1;4,12 (Fikkert, 1994; Levelt, 1994) and in English for e.g. G at 2;3-2;9 (Gnanadesikan, 1995), and P.J. at 1:11 (Demuth and Fee, 1995; Demuth, 1996). The data in 1) below show that this pattern was an intermediate stage for Trevor (Compton and Streeter, 1977; Pater, 1997), coming between his initial and final stage of coda acquisition:

* My thanks especially to Joe Pater, John McCarthy and Kathryn Flack, and also to Michael Becker, Karen Jesney, Shelley Velleman, Elliot Moreton, Elan Dresher, Jason Riggle, Doug Pulleyblank, Jill Beckman, and audiences at MOT 2006, the Brown/UMass 2006 workshop and WCCFL25 for ideas, alternatives and discussion. All the errors, I came up with myself. This work was in part funded by SSHRC Doctoral Grant #752-2004-0511.

1) Trevor's three stages of coda acquisition

	Singleton Codas			Complex Codas		
	Target	*Child*	*Age*	*Target*	*Child*	*Age*
a) *initial stage*	'duck'	[dʌ]	0;10.17	'plant'	[tɛ]	1;3.11
all codas deleted	'cup'	[kʌ]	1;1.0	'orange'	[oŋ]	1;4.2
(up to 1;4.2)	'puppet'	[pʌpə]	1;3.25			
b) *intermediate stage*	'walk'	[wɔk]	1;6.8	'box'	[gak]	1;7.11
singleton codas only	'hat'	[hæt]	1;6.8	'toast'	[toːs]	1;7.20
(1;5-1;7.26)	'melon'	[mɛːmɪn]	1;7.26	'milk'	[mʊːk]	1;7.26
c) *final stage*	'room'	[wuːm]	1;9.2	'plant'	[pænt]	1;9.2
all coda retained	'egg'	[ɛg]	1;9.28	'stairs'	[ʃtɛrʃ]	1;9.2
(1;9 onwards)	'outside'	[saiːd]	1;9.28	'toast'	[toːst]	1;9.29

Trevor's pattern in 1b) can be captured by the ranking in 2b) below. Compared to the initial stage of 2a), Trevor's grammar is one in which the general markedness constraint against all codas has been demoted, but the specific markedness constraint against complex codas remains high:

2) a) The initial stage: NoCoda, *ComplexCoda >> Max
b) A Specific Markedness stage: *ComplexCoda >> Max >> NoCoda

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

3) a) Singleton codas retained

/wɔk/	Max	NoCoda
☞ wɔk		*
wɔ	*!	

b) Coda clusters reduced

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

2.2 The Specific-F stage and the acquisition of French complex onsets

To illustrate the second kind of intermediate stage I use here some data from Rose (2000), who reports on a longitudinal study of Québécois French as acquired by two children, Clara and Théo. Both of these children pass through a stage at which complex onsets are preserved faithfully in stressed syllables, but the same clusters are reduced to singleton in unstressed syllables (shown here for Clara):

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

	Singleton Codas			Complex Codas		
	Target	*Child*	*Gloss*	*Target*	*Child*	*Gloss*
initial stage	/kχa.ˈkχa/	[ka.ˈkæ]	'Cracra'	/bʁiˈze/	[bœˈçiː]	'broken'
clusters reduced	/plœʁ/	[pœː]	'cries' (v)	/abʁiˈko/	[pupæˈko]	'apricot'
(1;0.28 - 1;09.01)	/flœʁ/	[βœː]	'flower'			
intermediate stage	/bi.ˈbʁɔ̃/	[pa.ˈpχɔ]	'bottle'	/fʁi.ˈgo/	[bʊ.ˈko]	'fridge'
only stressed	/glɪs/	[klɪs]	'slides' (v.)	/bʁy.ˈle/	[bɪ.ˈle]	'burned'
clusters retained	/si.ˈtχʊj/	[θə.ˈtχuːj]	'pumpkin'	/gli.ˈsad/	[ka.ˈsæd]	'slide'
(1;09.29-2;03.05)	/plœʁ/	[plœχ]	'cries' (v.)	/tχu.ˈve/	[tʊ.ˈve]	'found'
final stage	/ˈgʁo/	[ˈgʁo]	'big'	/tχu.ˈve/	[tχʊ.ˈve]	'found'
clusters retained				/ˈplɑ̃ʃe/	[ˈplɑ̃ʃe]	'floor'
(2;03.15)						

The ranking that I will use to capture this intermediate stage is in 5b) below. What has changed here compared to the initial stage is that a specific faithfulness constraint has been promoted:

5) a) The initial stage: *ComplexOnset >> Max-σ', Max-σ'
b) A Specific Faithfulness stage Max-(σ') >> *ComplexOnset >> Max

The top-ranked constraint in this ranking is a positional version of Max: its effect is to prevent input segments from being deleted from stressed syllables.[1] With markedness demoted below just this constraint, we get Clara's intermediate stage as below:

6) a) Clusters retained in stressed syllables

/gl̲ɪs/	Max-σ'	*CompOns
kl̲ɪs		*
k̲ɪs	*!	

b) Clusters reduced elsewhere

/gl̲i.ˈsad/	Max-σ'	*CompOns	Max
kl̲a.ˈsæd		*!	
k̲a.ˈsæd			*

See also Kehoe and Hilaire-Debove (2003) on another French-learning child's intermediate acquisition of some stop-glide clusters, and e.g. Lléo (2003) on coda cluster acquisition in Spanish.

2.3 *Summarizing the two intermediate stages*

So far I have characterized two developmental stages between the initial and final ones, using the two ranking schemas below:

7) a) The Specific-M stage: e.g. ***ComplexCoda** >> Max >> NoCoda
b) The Specific-F stage: e.g. **Max-σ'** >> *ComplexOnset >> Max

Note that the constraints I've used at the top and bottom strata of these rankings stand in *stringency* relations (see esp. Prince, 1997; de Lacy, 2002) – e.g. NoCoda is more stringent than *ComplexCoda because the former assigns a proper *superset* of the violations assigned by the latter. We will see the consequences of this stringency relation in section 5.

3. The theoretical starting point: Biased Constraint Demotion

My starting point is the recursive Biased Constraint Demotion algorithm (Prince and Tesar, 2004; see also the model of Hayes, 2004). BCD is an error-driven learning algorithm for classic OT, built to learn restrictively – that is, to avoid learning superset grammars (see Angluin, 1980; Berwick, 1982; Smolensky, 1996.) It excels at converging on the target grammar because it keeps track of all its errors in a table called the Support, and then reasons from the Support's data to necessary constraint rankings (using the logic of Tesar and Smolensky, 2000). While it is efficient and restrictive, BCD is not an algorithm built to model realistic human learning – thus, for example, it is not designed to explain why children go through intermediate stages.

3.1 *How and why BCD learns too fast*

To get from one ranking to the next, the BCD learner uses their current grammars to make errors – that is, they submit outputs in the ambient language to their current EVAL, and note each time their ranking produces something different. The erroneous outputs (called *losers*) are analyzed by comparing them to their ambient forms (called *winners*) and considering the role of each constraint in preferring each winner or loser. These comparative preferences are what the BCD learner then stores in the Support:

---

[1] Adopting this constraint requires the assumption, which I do not have room here to discuss, that children's inputs have syllabic structure. This means in Clara's case that her grammar must syllabify the stressed clusters that she retains at the intermediate stage as complex onsets, and not coda-onset clusters, e.g. [pa.ˈp̲χɔ], *[pap̲.ˈχɔ].

8) How the ‘pure’ BCD learner gets errors into the Support

a) An error is made…

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

b) … and is added to the Support table

Winner ~ Loser	NoCoda	*CompCoda	Max
tost ~ to	L	L	W

Having added this error to the Support, the BCD algorithm now builds a new ranking that will prefer the winnner over the loser (and will therefore better approximate the final grammar.) Space constraints do not permit me to do any real justice to this approach here, and readers unfamiliar with BCD should consult Prince and Tesar (2004), as well as both Tesar and Smolensky (2000) and Hayes (2004). But in this simplified case, the reasoning that BCD uses to build that ranking can be paraphrased as follows: Why did the current grammar made this error? Because two constraints that are ranked too high preferred the losers: these are NoCoda and *ComplexCoda. So how can we prevent this error from being made in the new grammar? By demoting these two constraints below a constraint that prefers the winners: this is Max. Thus the learner arrives at the ranking in 9) below:

9) The grammar that BCD learns from 8b)’s Support:
Max >> NoCoda, *ComplexCoda

10) This ranking’s effect: all codas retained, including complex ones

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

In short: a pure BCD learner who makes the error in 8a) on ‘toast’ will immediately learn the correct ranking in 9) above. So if we want our learner to learn slower, it will have to either not learn from this error, or not learn using BCD. As we will now see, Error-Selective Learning adopts the first approach: it exploits the success of BCD at getting from errors to rankings, but enriches the mechanism with which it chooses those errors.

4. The proposal: using Error Selection to feed BCD

The error-selective approach in a nutshell is as follows. At the onset of each new stage of development, the learner uses their the current grammar for a while and accumulates a body of errors. Eventually, the learner chooses from that amassed data *just one error* to learn from – an error which will force the learner to change as few of their current grammar’s rankings as possible. This error is added to the Support, a new grammar is built using the BCD algolrithm, and a new stage begins.

4.1 Walking through Error-Selective learning: Trevor’s complex codas

The first part of the gradualness of error-selective learning is its slow reaction to errors. When the Error-Selective learner makes an error, it is not immediately added to the Support; nor does it trigger re-ranking on its own. Instead, errors are filed in a temporary storage area called the *Error Cache*. The Cache holds all errors made by the current grammar hypothesis, analyzed but unresolved:

11) <u>How the Error-Selective Learner adds errors to the Error Cache</u>

a) Trevor's grammar makes this error…

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

b) … he adds it to the current Error Cache (already including other errors)…

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

c) … but the Support table is **NOT** updated:

Winner ~Loser	NoCoda	*CompCoda	Max
… empty, waiting…			

Adding an error to the Cache has no other immediate effect. Errors continue to be made with the current grammar, and so over time the Cache grows. In other words: although each error in the Cache provides evidence of a new, better grammar that BCD could build, the error-selective learner waits.

So when *is* this gradual learner finally forced to acquire a new grammar? Learning occurs when a constraint overcomes the *Violation Threshold* – that is: when a constraint has assigned Ls to more than some specified number of errors in the Error Cache. In ESL, that constraint is called the *Trigger Constraint,* because it has triggered learning.

Let us imagine that Trevor is a very quick learner, and so his violation threshold is 3. This means that Trevor will learn a new ranking every time some constraint has assigned an L to 3 different winner~loser pairs. Such a Cache is in 12) below:[2]

12) a sample Error Cache, in which NoCoda has triggered learning:

Winner ~Loser	***NoCoda***	**CompCoda*	*Max*	**CompOnset*
(i) tost ~ to	**L**	L	W	e
(ii) piz ~ pi	**L**	e	W	e
(iii) greʲp ~ ge	**L**	e	W	L
(iv) ti ~ si	e	e	e	e

Now that learning has been triggered, Trevor must choose one of the errors in this Cache – what I'll call the *Best Error* – to add to his Support. This is done via an Error-Selection Algorithm, which can be simply defined here as:

13) <u>The Error Selection Algorithm (first pass)</u>
Choose as the Best Error some row in the Cache which:
a) has an L assigned by the Trigger Constraint
b) has the fewest Ls assigned by *other* Markedness constraints

For Trevor's Cache in 12), these two criteria choose a unique Best Error. Criterion (a) eliminates 'tea', because it does not have a NoCoda violation, and criterion (b) eliminates 'toast and 'grape', because they also have Ls assigned by other markedness constraints.

[2] The Violation Threshold is just a numerical value chosen for modeling purposes – and clearly in any realistic learning model, the Threshold would have to be set much, much higher than 3.

Thus the Best Error in this Cache is 12ii): the error made on ‘peas’. Having determined this, Trevor adds 12ii) to the Support and clears all other errors from his Cache. This process is schematized in the diagram in 14) below.

14) How the Error-Selective Learner adds errors to the Support

a) Trevor’s grammar makes this error, and adds it to the Cache…

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

b) **NoCoda becomes the Trigger Constraint…**

Winner ~Loser	*No Coda*	**Comp Coda*	*Max*	**Comp Onset*
tost ~ to	**L**	L	W	e
piz ~ pi	**L**	**e**	**W**	**e**
greʲp ~ ge	**L**	e	W	L
ti ~ si	**e**	e	e	e

c) the pre-existing Support table:

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

d) … the Cache gets cleared

Winner ~Loser	*No Coda*	**Comp Coda*	*Max*	**Comp Onset*
… empty, waiting to start again…				

e) **… ‘peas’ is added to the Support:**

W ~ L	No Coda	*Comp Coda	Max	*Comp Onset
piz ~ pi	L	e	W	e

Now that the Support has been updated in 14e), the BCD algorithm can be applied as normal to reason from the Support’s errors to a new ranking. Since the ESA made Trevor choose an error like ‘peas’ with a *singleton* coda, his Support only contains evidence that singleton codas need to be tolerated in the new grammar. This means that NoCoda will be re-ranked below Max. The biases of Biased Constraint Demotion continue to impose the M >> F ranking bias at every stage – so all other markedness constraints will remain high-ranking in Trevor’s new grammar, including *ComplexCoda:

15) The intermediate stage ranking that BCD builds from 14e)’s Support: (see 4)
*ComplexCoda >> Max >> NoCoda

And since Trevor has cleared his Cache, his previous errors on other words with other, more marked structures – i.e. the complex onset of ‘grape’ and the complex coda of ‘toast’ – are now forgotten. All that he retains of his previous stage is the one surviving Best Error, which built him the ranking in 15). In this way, Trevor has entered his intermediate Specific-M stage (§2.1).

4.2 A second example: learning Clara’s complex onsets

In §2.2 we saw an example of the Specific-F stage, from Clara’s acquisition of complex onsets in Québécois French. Recall that at her initial stage, complex onsets are reduced to singletons in all contexts, while at the intermediate stage they are retained in stressed syllables only, and still reduced elsewhere. How can Error-Selective learning derive this kind of intermediate stage? To begin, we will consider the point at which Clara accumulated enough errors for *ComplexOnset to overcome the Violation Threshold, and so had to choose an error to add to her Support. The two criteria of the error selection algorithm in 13) choose one that violates *ComplexOnset and as few other Markedness constraints as possible – suppose that in Clara’s Cache at the end of the initial state, this narrowed the choice to two candidates:

16) a sample of Clara's potential Best Errors:

Winner ~Loser	***CompOnset**	*NoCoda*	**front/rd-V*	*Max*	*Max-σ'*
(i) ˈplœʁ ~ pœː	**L**	L	e	W	W
(ii)b̲ʁy.ˈle ~ b̲ɪ.ˈle[3]	**L**	e	L	W	e

Since the goal of ESL is to ensure the learner changes their grammar minimally at each stage, in a way consistent with data like Clara's, I add the following third criterion to the error-selection process:

17) The Error Selection Algorithm (full version)
Choose as the Best Error some row in the Cache which:
a) has an L assigned by the Trigger Constraint
b) has the fewest Ls assigned by *other* Markedness constraints, and among those, one which
c) has the most Ws assigned by Faithfulness constraints

Looking at Clara's two errors above, we can see that this faithfulness criterion will choose the error in 16i), that has Ws assigned by *both* the general Max constraint *and* the specific Max-σ' constraint.

18) Clara's Support, containing the best error chosen from the Cache in 16)

Winner ~ Loser	*ComplexOnset	Max	Max-σ'
ˈplœʁ ~ pœː	L	W	W

To see how this error will cause minimal learning, we need to know the full set of biases that our Biased Constraint Demotion will enforce when constructing a ranking. When forced to choose a faithfulness constraint to install, the BCD algorithm that I adopt picks the *most specific faithfulness constraint that prefers winners* (see Hayes, 2004; Tessier, 2006.[4]) Thus adding an error like 18) to the Support will allow this BCD algorithm to install just the specific Max-σ' above *ComplexOnset, producing the intermediate stage:

19) The resulting ranking that the Specific-F bias makes BCD learn from 19)'s Support (see 6):
Max-σ' >> *ComplexOnset >> Max

5. Some discussion of Error-Selective Learning

As presented above, the goal of Error-Selective Learning is to derive the kind of stages in the first section: those that that rely crucially on demoting general markedness before specific, and promoting specific faithfulness before general. We can now see how the ESL approach achieves this gradualness in learning. On the one hand, using a Violation Threshold to trigger learning means that more stringent Markedness constraints will trigger learning before less stringent (more specific) ones – because e.g. NoCoda will always assign Ls to all the errors that *ComplexCoda does, and then some. Combining the Threshold approach with the first two criteria of error selection in 17) derives Specific-M rankings: this learner choose errors that will demote the *general* Trigger Constraint like NoCoda but as few other constraints (like *ComplexCoda) as possible.

On the other hand: since more stringent constraint like Max assigns Ws to every error that specific Max-σ' does, and then some, errors that have the most faithfulness Ws will be those with marked structure in the contexts of specific faith. This means that the third error-selection criterion will choose errors with Ws assigned by specific faith constraints -- and this will derive Specific-F rankings, because those errors will cause the present BCD algorithm to promote only the *specific* faithfulness constraints like Max-σ'.

[3] Note that this error is hypothetical: Rose only provides a pronunciation for 'bruler' at the interemdiate stage.
[4] Cf. Prince and Tesar (2004), who do not adopt the Specific-F bias in choosing among installable faithfulness constraints – for good reasons which are addressed by both Hayes and Tessier.

Beyond specific-to-general constraint relations, Error-selective Learning also makes general predictions about the order of acquisition of marked structures. All other things being equal, the order of acquisition of marked structure should correlate with the frequency of that structure in the target outputs that are causing errors – a prediction that appears to be larger correct (see e.g. the results reported in Roark and Demuth, 2000; Kirk and Demuth, 2003.)

While the Error-Selective learner forgets many of the errors it makes, by continually emptying the Cache, it is important to note that ESL will still converge on a final, restrictive grammar. It will do so for the same reason that pure Biased Constraint Demotion does – namely, because it always builds a ranking from scratch, using the sum of its errors in the Support. Every time a constraint exceeds the Violation Threshold, some new error is chosen to add to the Support. From then on, the BCD algorithm will always build grammars with that error in mind; and given BCD's efficiency, this error will never made again. Because the Cache is emptied every time a new grammar has been built, the next cycle of ESL will always teach the learner something new about the target grammar. Eventually all the errors necessary to finding the correct grammar will have been added to the Support, the learner will apply BCD one last time, and they will settle on a grammar that makes no more errors. This ranking will thus be their final grammar: one that can account for all the data they've seen but which otherwise remains restrictive.

The Error-Selective BCD learner is an alternative to existing OT learning models that produce intermediate stages, which do so by encoding probabilistic information directly into the grammar – most notably the Gradual Learning Algorithm (Boersma, 1998; Boersma and Hayes, 2001; Curtin and Zuraw, 2001.) In part, the current approach is therefore proposed to remove one of the biggest arguments against using a classic view of OT to model natural language learning.

6. Summary and conclusions

This paper has introduced a gradual method of learning OT grammars via Biased Constraint Demotion. In the Error-Selective proposal, learners slowly process the errors that will lead them to their target rankings, and only search for a new grammar when a number of errors suggest the same problem with their current one. This learner relies on comparative error data in the Support to both choose errors to learn from (via the error-selection algorithm in 17) and build new rankings (via a somewhat revised version of BCD.)

We have seen above how Error-Selective BCD learning can derive some common intermediate stages in phonological acquisition. It is hoped that this result demonstrates one way in which the insights from learnability research – namely the search for restrictive constraint rankings that drives BCD – can be incorporated into a model of natural language learning, in all its gradual and imperfect glory.

If we adopt as our ultimate goal a phonological learning model that is truly realistic and yet restrictive, there are of course a panoply of broader questions that have been left unexplored here (though see Tessier, 2006 for some further discussion.) One family of questions are about the extent to which this approach can derive a fuller range of developmental stages, including the variation between stages, grammatical regressions, and so forth. Another cluster of questions concern the precise roles of frequency in guiding or determining phonological development. For example: what linguistics objects should frequency be calculated across? children's own productions or lexicons of child-directed speech? types or tokens? Such issues remain interesting avenues of further research, with which to test the overall success of the Error-Selective learner.

References

Angluin, Dana (1980). Inductive inference of formal languages from positive data. *Information and Control* vol. 45. pp. 117-135.

Berwick, Robert (1982). *Locality principles and the acquisition of syntactic knowledge*. PhD dissertation, MIT.

Boersma, Paul (1997). *Functional Phonology: Formalizing the interactions between articulatory and perceptual drives*. Ph.D. dissertation, University of Amsterdam. The Hague: Holland Academic Graphics.

Boersma, Paul and Bruce Hayes (2001). Empirical Tests of the Gradual Learning Algorithm. *Linguistic Inquiry vol. 32*. pp 45-86.

Compton, A.J. and M. Streeter (1977). Child Phonology: Data Collection and Preliminary Analyses. *Papers and Reports on Child Language Development 7.* Stanford, CA: Stanford University.

Curtin, Suzanne and Kie Zuraw (2001). In Barbora Skarabela, Sarah Fish, and Anna H.-J. Do (eds.), *Proceedings of BUCLD 26.* pp. 118-129. Somerville, MA: Cascadilla Press.

de Lacy, Paul V. (2002). *The Formal Expression of Markedness.* Ph.D. dissertation, UMass Amherst.

Demuth, Katherine, & E. Jane Fee. 1995. Minimal Words in Early Phonological Development. Ms., Brown University and Dalhousie University.

Demuth, Katherine (1996). Stages in the acquisition of prosodic structure. In E. Clark (ed.), *Proceedings of the 27th Child Language Research Forum*. pp. 39-48. Stanford University: CSLI.

Fikkert, Paula (1994). *On the Acquisition of Prosodic Structure*. Ph.D. dissertation, University of Leiden.

Gnanadesikan, Amahlia (1995/2004). Markedness and Faithfulness Constraints in Child Phonology. In Kager, Rene, Joe Pater & Wim Zonneveld (eds.), *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge, U.K.: Cambridge University Press.

Hayes, Bruce (2004). Phonological Acquisition in Optimality Theory: the early stages. In Kager, Rene, Joe Pater & Wim Zonneveld (eds.), *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge, U.K.: Cambridge University Press.

Jakobson, Roman (1941). *Kindersprache, Aphasie, und algemeine Lautgesetze*. Uppsala: Almqvist & Wiksell [tr. Child Language, Aphasia and Phonological Universals, Mouton].

Kehoe, Margaret and Geraldine Hilaire-Debove (2003). The Structure of Branching Onsets and Rising Diphthongs: Evidence from the Acquisition of French. In Alejna Brugos, Linnea Micciulla, and Christine E. Smith (eds.), *Proceedings of BUCLD 28.* pp. 282-293. Somerville, MA: Cascadilla Press.

Kirk, Cecilia and Katherine Demuth (2003). Onset/Coda Asymmetries in the Acquisition of Clusters. In Barbara Beachley, Amanda Brown, and Frances Conlin (eds.), *Proceedings of BUCLD 27.* pp. 437-448. Somerville, MA: Cascadilla Press.

Levelt, Claartje (1994). *On the Acquisition of Place.* Ph.D. dissertation Leiden University, HIL Dissertation Series 8.

Lléo, Conxita (2003). Prosodic licensing of codas in the acquisition of Spanish. *Probus* vol. 15. pp 257-281.

Pater, Joe (1997). Minimal Violation and Phonological Development. *Language Acquisition* vol. 6(3). pp 201-253.

Prince, Alan (1997). Stringency and anti-Paninian hierarchies. Handout from LSA Summer Institute.

Prince, Alan and Bruce Tesar (2004). Learning Phonotactic Distributions. In Kager, Rene, Joe Pater & Wim Zonneveld (eds.), *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge, U.K.: Cambridge University Press.

Roark, Brian and Katherine Demuth (2000). Prosodic Constraints and the Learner's Environment: a Corpus Study. In S. Catherine Howell, Sarah A. Fish, and Thea Keith-Lucas (eds.), *Proceedings of BUCLD 24.* pp. 597-608. Somerville, MA: Cascadilla Press.

Rose, Yvan (2000). Headedness and Prosodic Licensing in the L1 Acquisition of Phonology. Ph.D. dissertation, McGill University.

Smolensky, Paul (1996). The initial state and 'Richness of the Base.' Technical Report JHUCogSci-96-4. ROA-154.

Stampe, David (1969). The acquisition of phonetic representation. In *Papers from the Fifth Regional Meeting, CLS*. pp 443-454. Chicago: Chicago Linguistic Society.

Smith, Neilson V. (1973). *The Acquisition of Phonology: A Case Study.* Cambridge, MA: Cambridge University Press.

Tesar, Bruce and Paul Smolensky (2000). *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.

Tessier, Anne-Michelle (2006). *Biases and Stages in Phonological Acquisition.* Ph.D. dissertation, UMass Amherst.

Proceedings of the 25th West Coast Conference on Formal Linguistics

edited by Donald Baumer, David Montero, and Michael Scanlon

Cascadilla Proceedings Project Somerville, MA 2006

Copyright information

ISBN 1-57473-415-6 library binding

Ordering information

Orders for the library binding edition are handled by Cascadilla Press.
To place an order, go to www.lingref.com or contact:

Cascadilla Press, P.O. Box 440355, Somerville, MA 02144, USA
phone: 1-617-776-2370, fax: 1-617-776-2271, e-mail: sales@cascadilla.com

Web access and citation information

This entire proceedings can also be viewed on the web at www.lingref.com. Each paper has a unique document # which can be added to citations to facilitate access. The document # should not replace the full citation.

This paper can be cited as:

Tessier, Anne-Michelle. 2006. Stages of Phonological Acquisition and Error-Selective Learning. In *Proceedings of the 25th West Coast Conference on Formal Linguistics*, ed. Donald Baumer, David Montero, and Michael Scanlon, 408-416. Somerville, MA: Cascadilla Proceedings Project.

or:

Tessier, Anne-Michelle. 2006. Stages of Phonological Acquisition and Error-Selective Learning. In *Proceedings of the 25th West Coast Conference on Formal Linguistics*, ed. Donald Baumer, David Montero, and Michael Scanlon, 408-416. Somerville, MA: Cascadilla Proceedings Project. www.lingref.com, document #1474.