Week 5 – Learning OT Grammars

March 17 and 19, 2008

1 Acquisition and learnability: brief intro

(1) The "Acquisition"/ "Learnability" Distinction

a. "Acquisition": examine the behavior (both production and perception) of infants and children to try to figure out how they are doing it.

b. "Learnability": develop formal models of how language could be acquired given UG and data like what infants and children hear.

(2) Learnability studies: **what is needed**

a. learning-centric linguistic theories (cf. Hayes and Wilson's (2008) propaganda term: "learning-theoretic phonology")

b. ingenious methods to test whether a proposed learning system is learning the right thing; these in turn will rely on

(i) detailed knowledge of the language(s) learned

(ii) experimental skills and ingenuity

(3) Learnability in OT: Some Tasks

a. Constraints: if UG does not provide them, learn them.

b. Phonotactics: learn what is legal

c. Morphology and Alternation

(i) Parse the words into their component morphemes, find morphemes that alternate, deduce their underlying form.

(ii) Develop a morphological component (Optimality-theoretic?) that determines what words can be constructed and in what way (morpheme order, etc.)3

(iii) Find a constraint ranking that derives all the surface forms correctly from the underlying forms and given constraints. [*our focus here*]

2 The Constraint Demotion Algorithm for ranking OT constraints

(4) Recursive Constraint Demotion (RCD) (Tesar 1995, Tesar and Smolensky 1998, 2000).

a. See also: `http://ruccs.rutgers.edu/roa.html`
   (i) Tesar, Bruce and Paul Smolensky (1993) “The learning of Optimality Theory: An algorithm and some basic complexity results,” ROA-52.
   (ii) Tesar, Bruce and Paul Smolensky (1996) “The learnability of Optimality Theory,” (much longer, with ideas about accessing underlying forms) ROA-156.

(5) Problem addressed: constraint ranking for morphophonemics, given underlying forms.

Algorithm 1 Constraint Demotion in plain language

Input: a set of ERCs, $n = 0$
Output: a stratified constraint hierarchy (if there is one)
while There are ERCs remaining **do**
  1. Add one to n
  2. Inspect the complete set of pairs consisting of a winner and a losing contender for the same input. Among these, find the set of constraints $\{C_1, C_2, \ldots C_n\}$ that are never “prefer a loser”.
  3. These can safely be ranked in the current stratum. So let $\{C_1, C_2, \ldots C_n\}$ constitute level n in the strata of constraints.
  4. Throw away rival candidates that are already explained; i.e., which violate at least one of $\{C_1, C_2, \ldots C_n\}$ more than the winner does. This means that in the culling process, they are guaranteed to be culled before the winner is.
  5. Repeat the while loop ad lib, thereby forming successively lower strata, until all the data are explained.
end while

2.1 Entailed Ranking Conditions (ERCs)

(6) Reviewing the crucial aspect of OT: winner selection
   a. Going through the constraints in ranked order, winnow out any candidate that violates the current constraint more times than any other candidate.

(7) Another way to represent the information in a tableaux is to use ERCs (Prince 2002).

ERC: a vector made up of elements from the set {W,L,e}

(8) ERCs:
   a. For each (winner,contender) pair there is an ERC.
   b. The length of the ERC is the same as the number of constraints.
   c. Each cell in the vector tells you whether the constraint indexed to that position
      (i) prefers the winner (W)
      (ii) prefers the losing contender (L)
      (iii) prefers neither the winner or losing contender (e)

(9) Example: <W,W,W,L,e> means

a. Constraint 1 prefers the winner.
b. Constraint 2 prefers the winner.
c. Constraint 3 prefers the winner.
d. Constraint 4 prefers the loser.
e. Constraint 5 does not prefer the winner or loser.

(10) Standard tableaux can be converted into a set of ERCs. Suppose you had the following constraints ranked as follows:

/bad/		NoCoda	$_\sigma$[CC	Max(C)	Dep(V)	Linearity
a. ☞	badi				*	
b.	bad	*				
c.	ba			*		
d.	bda		*			*
e.	baid	*			*	

(11) Conversion procedure:

a. Compare the winner candidate with another candidate *pointwise*. For each constraint C:
  (i) If the other candidate has more violations of C than the winner, fill in 'W'. ("C prefers the winner")
  (ii) If the winner has more violations of C than the other candidate, fill in 'L'. ("C prefers the loser")
  (iii) If they have the same violations, fill in 'e'. ("C doesn't care").

(12) ERC tableaux (also called comparative tableaux):

/bad/		NoCoda	$_\sigma$[CC	Max(C)	Dep(V)	Linearity
a∼b.	badi∼bad	W	e	e	L	e
a∼c.	badi∼ba	e	e	W	L	e
a∼d.	badi∼bda	e	W	e	L	W
a∼e.	badi∼baid	W	e	e	e	e

(13) Quick things to remember:

a. If an ERC contains no Ls then you have identified a harmonically-bounded candidate.
b. If an ERC contains no Ws, then you every constraint has no preference or prefers the loser! I.e. no ranking will work, your 'winner' candidate is harmonically-bounded!
c. If an erc contains one W and one L, you have a ranking argument.

★ Explain in words what the erc <W,e,e,L,e> means. What about <e,W,e,L,W>?

(14) Let's convert a tableaux you may have used in the Ilokano homework.

Assume no free variation because it makes things complicated.

Constraint Name	Abbr.	Comment
Max-IO(V)	Max(V)	Undominated
Onset	Onset	Undominated, the basis of the conspiracy.
*[-cons, -syl, +low]	*Low Glide	Keeps /a/ from gliding while staying low.
*[-cons, -syl, -high]	*Nonhigh Glide	Forces /e/ to glide to [j], not [e̯]
Ident-IO(low)	ID(low)	Keep /a/ from gliding by raising, e.g. to *[w]
Dep-IO (ʔ)	Dep(ʔ	violable, since epenthesis of ʔ can happen
Ident-IO (high)	ID(high)	Violated in winners: /e/ → j, /o/ → w; must be ranked low
Ident-IO([syllabic])	ID(syl)	Violated in all winners with gliding; must be ranked low

(15) Convert the following tableaux to an ERC tableaux.

/basa-en/	Max(V)	Onset	*LowGlide	*NonHigh	ID(low)	Dep(ʔ	ID(high)	ID(syl)
a. ☞ ba.sa.ʔen						*		
b. bas.wen					*			
c. bas.a̯en			*	*				
d. ba.sa.en		*						
e. ba.san	*							

(16) Erc Tableaux

/basa-en/	Max(V)	Onset	*LowGlide	*NonHigh	ID(low)	Dep(ʔ	ID(high)	ID(syl)

2.2 Erc Fusion

(17) Prince (2002) introduces an operation called 'fusion' (∘) which lets us combine the information in ercs.

∘	W	L	e
W	W	L	W
L	L	L	L
e	W	L	e

(18) Principles of erc fusion

a. 'L extension' : L $\circ x$ = L (for any x)
b. 'W retraction' : W $\circ x$ = W iff x = W or x = e
c. 'e' identity : e $\circ x = x$.

(19) This is applied pointwise to ercs. So:

a. <W,L,L,e> ∘ <L,W,L,W> = <L,L,L,W>
b. <e,e,e,W,L> ∘ <W,L,e,e,e> = <W,L,e,W,L>

★ Explain in words what these three ercs (19-b) mean.

<e,	e,	e,	W,	L>
<W,	L,	e,	e,	e>
<W,	L,	e,	W,	L>

(20) Erc fusion loses information, but doesn't lose all the relevant information!

★ Fuse all the ercs in (16). What do you know from the fused erc?

2.3 Basic procedure for Recursive Constraint Demotion

(21) Constraint Strata

a. Constraint strata are partial orderings of the constraints. The above algorithm is essentially what OT does to produce hierarchies.
   (i) If C1 is in higher stratum than C2, then C1 is assumed to outrank C2 (Assume level 1 is at the top).
   (ii) If C1 and C2 are in the same stratum, then their ranking is asserted not to matter; pick any ranking you like and it will yield the same result. Thus, a set of strata defines a (possibly large) set of complete rankings, any of which are claimed to work.

Algorithm 2 Constraint Demotion in terms of ERCs

Input: a set of ERCs, $n = 0$
Output: a stratified constraint hierarchy (if there is one)
while There are ERCs remaining **do**
1. Add one to n
2. Find all those constraints that never prefer losers. This can be done by 'erc fusion'.
3. Remove from the erc-set those ercs generated by winner-loser pairs. These are the candidates 'handled' by the found constraints.
4. Shorten the remaining ERCs by removing the cells which correspond to the found constraints. This remaining set is passed recursively through the while loop.
5. The found constraints constitute Level n in the constraint strata.
end while

(22) Proof of correctness (Tesar 1995, Tesar and Smolensky 2000).

a. Provided you have all the "contenders" for every input, as defined by the constraint set, constraint demotion is guaranteed to find a consistent ranking, if there is one.

(23) The algorithm correctly crashes when there is no ranking that yields the desired outputs. At some stage: it finds that every constraint still under examination favors a loser (fused ercs yields an erc with no Ws).

2.4 Applying Constraint Demotion to Ilokano

(24) First we have to collect all of our ERCs.

(25) Here is the the tableaux we converted earlier.

/basa-en/	Max(V)	Onset	*LowGlide	*NonHigh	Id(low)	Dep(ʔ	Id(high)	Id(syl)
a. ☞ ba.sa.ʔen						*		
b. bas.wen					*			
c. bas.a̯en			*	*				
d. ba.sa.en		*						
e. ba.san	*							

/basa-en/		MAX(V)	ONSET	*LOWGLIDE	*NONHIGH	ID(LOW)	DEP(ʔ	ID(HIGH)	ID(SYL)
a∼b.	ba.sa.ʔen∼bas.wen	e	e	e	e	W	L	e	e
a∼c.	ba.sa.ʔen∼bas.a̯en	e	e	W	W	e	L	e	e
a∼d.	ba.sa.ʔen∼ba.sa.en	e	W	e	e	e	L	e	e
a∼e.	ba.sa.ʔen∼ba.san	W	e	e	e	e	L	e	e

(26) Here is another tableaux to convert to ERCs.

/babawi-en/		MAX(V)	ONSET	*LOWGLIDE	*NONHIGH	ID(LOW)	DEP(ʔ	ID(HIGH)	ID(SYL)
a. ☞	ba.baw.jen								*
b.	ba.ba.wi.ʔen						*		
c.	ba.ba.wi.en		*						
d.	ba.ba.win	*							

/babawi-en/	MAX(V)	ONSET	*LOWGLIDE	*NONHIGH	ID(LOW)	DEP(ʔ	ID(HIGH)	ID(SYL)

(27) And another:

/masahe-an/		MAX(V)	ONSET	*LOWGLIDE	*NONHIGH	ID(LOW)	DEP(ʔ	ID(HIGH)	ID(SYL)
a. ☞	ma.sah.jan								*
b.	ma.sa.he.ʔan						*		
c.	ma.sah.e̯an				*				
d.	ma.sa.he.an		*						
e.	ma.sa.han	*							

/masahe-an/	MAX(V)	ONSET	*LOWGLIDE	*NONHIGH	ID(LOW)	DEP(ʔ	ID(HIGH)	ID(SYL)

(28) Now let's put all these ercs into one big set, and we can begin the algorithm.

MAX(V)	ONSET	*LOWGLIDE	*NONHIGH	ID(LOW)	DEP(ʔ	ID(HIGH)	ID(SYL)

(29) Let's apply the algorithm!

a. **Step 1.** Add one to n.
   (i) OK, so n=1.

b. **Step 2.** Find all those constraints that never prefer losers with 'erc fusion'.
   (i) Fuse all the ercs in (28). What do we get?
   (ii) Which constraints never prefer losers? They form the first stratum!

c. **Step 3.** Remove from the erc-set those ercs generated by winner-loser pairs. These are the candidates 'handled' by the found constraints.

   ★What is our remaining erc set?

d. **Step 4.** Shorten the remaining ERCs by removing the cells which correspond to the found constraints. This remaining set is passed recursively through the while loop.

★What is does the erc set look like now?

(30) Now we have to repeat these steps again as our erc set is still not empty.

- a. **Step 1.** Add one to n.
  - (i) OK, so n=2.
- b. **Step 2.** Find all those constraints that never prefer with 'erc fusion'.
  - (i) Fuse all the ercs in (28). What do we get?
  - (ii) Which constraints never prefer losers? They form the second stratum!
- c. **Step 3.** Remove from the erc-set those ercs generated by winner-loser pairs. These are the candidates 'handled' by the found constraints.

★What is our remaining erc set?

- d. **Step 4.** Shorten the remaining ERCs by removing the cells which correspond to the found constraints. This remaining set is passed recursively through the while loop.

★What is does the erc set look like now?

3 Learning phonotactics

(31)
- a. As we have seen, OT can analyze phonotactics (in isolation from alternations).
- b. However, Constraint Demotion will not suffice for learning such OT grammar.

(32)
- a. Ambient data: [bk], [bk], [blnd]
- b. Constraints: Max(C), *stop-liquid onset, *stop-nasal onset
- c. Task: learn [blık], *[bnık]

★ What is the ranking needed? How could Constraint Demotion find it?

(33) The general problem

- a. Constraint Demotion puts constraints about which it knows nothing as high as it can, and that includes, usually, Faithfulness contraints. They go low only when violated by winners; e.g. cat[s], dog[z].
- b. This produces a bad phonotactic grammar.

(34) Note that if English had alternations like these:

a-bla	bla
a-bna	bəna

we'd be much better off! But most phonotactic patterns do not "reveal" themselves this way.

a. In the absence of evidence from alternations, we assume that underlying forms are identical to surface forms. Hence no Faithfulness constraint ever prefers a loser, under these circumstances.

(35) So the task of raw phonotactic learning is beyond Constraint Demotion.

(36) More generally, we have the classical problem of learning when no evidence about ill-formedness is directly given to the learner.

(37) Ways in which the Constraint Demotion approach to learning is unsatisfying:

a. gradient phonotactics
b. input sample (UR,SR) pairs
c. Where is phonology in Constraint Demotion?

(38) Gradient phonotactic judgments (From Albright and Hayes (2003); averaged rating of 20 consultants in the frame "I like to ____ "):

[wɪs, kɪp]	5.84 on a 1-7 scale
[skɪk]	4.00
[snɔɪks]	3.00
[θwiːks]	2.53
[smiːnθ]	2.06
[bzɑrʃk]	1.50

(39) All similar experiments have yielded similar results.

(40) See Prince and Tesar (2004), Hayes (2004) for attempts to beef up Constraint Demotion (e.g. Biased Constraint Demotion).

(41) There are also Non-OT (but constraint-based) approaches (Heinz 2007, Hayes and Wilson 2008)

(42) Summary so far: Constraint Demotion is an effective, fast algorithm exists to rank constraints given (UR,SR) pairs and alternations. Junior, aged 8 months, probably knows very little morphology, and hence very few underlying forms—and yet seems to be making progress; see below.

4 Acquisition

(43) There's going to be a class on this next Fall, so this is a just a brief survey of some main results.

(44) Two strands of research

a. Older: diary and corpus study of child production: the (relatively) systematic set of mutilations that toddlers inflict on the adult language. The classic work here is Neilson Smith (1973) The Acquisition of Phonology; there is a very large literature

b. More recent: experimental work probing what children (passively) know. This includes infants.

4.1 Passive knowledge is far ahead of perception

(45) Perception work on infants: Increasing experimental evidence indicates that 10-month-olds have remarkable phonotactic knowledge. (blick vs. *bnick). See Jusczyk et al. (1993) and Friederici and Wessels (1993). For literature review and discussion, see Hayes (2004)

(46) The crucial experiments

a. Legal and illegal syllables are played to babies from one of two speakers.

b. A monitor, who for objectivity can't hear the syllables, times how long the babies turn their heads toward the speaker.

c. Babies like familiar syllables, not the novel, phonotactically-odd ones.

d. Strikingly, they even like more frequent syllables (Jusczyk et al.)—perhaps they haven't heard enough rare syllables to know for sure that they are possible.

(47) Perception leads production II: what little children say (Jusczyk et al. 1993, Friederici and Wessels 1993, Smith 1973)

(48) Smith (1973). Smith's Evidence:

a. Children can make use of distinctions they can't say (mouse/mouth).

b. Forms previously neutralized all become unneutralized at once: the child knew the correct form all along, and become able to say it. Smith, p. 139:

> Once [Smith's son] had learnt to produce clusters of a consonant plus [l] ... for both of adult /Cl/ and /Cr/, this cluster appeared immediately and correctly in words which it is quite certain he had not heard since before the critical day:
>
> *ground* → [glaʊnd] (previously [g̥aʊnd]) *footprint* → [wʊtplɪt] etc.
>
> quite spontaneously.

(49) Children recognize adults' imitations of their speech as such.

(50) Consequences

a. Unlike with adults, you can't tell what kids know by listening to what they say.

b. Baby talk is unlike adult talk because toddlers mutilate in their pronunciation what they (more or less accurately) know.

c. Research in child phonology is undergoing a shift from production studies to perception studies (syntax too, as well).

(51) Theoretical interpretation: how many phonologies does a young child have?

a. **One.** See Smolensky (1996). I.e. one grammar, that can be accessed both in perception and for the child's own productions. This work does not take into account the ability of children to pass the "blick" test.

b. **Two.** Menn (1983); Hayes (2004), etc.: The child "invents a personal phonology," for purposes of controlling clumsy childish articulators. This phonology maps adult surface forms to child surface forms. The child also has a "real" phonology, which represents her tacit knowledge of the adult language, and with which she passes the "blick" test.

c. **One point five.** See Hale and Reiss (1998). For Hale and Reiss, the child's own productions are not due to any kind of cognitive system at all, but are merely physiological, "the product of the body."

4.2 Consequences for research I

(52) It is plausible to attack the problem of learning phonotactics as a quasi-isolated problem, as in the algorithms mentioned above. No research of which I'm yet aware has stacked up phonotactic learning algorithms against infant perception data.

(53) Children's production phonologies and OT

a. Gnanadesikan, Amalia (1995) "Markedness and Faithfulness constraints in child phonology," ROA 75.

b. Pater, Joe (1996) "Minimal violation and phonological development", ms., Department of Linguistics, University of Alberta, Edmonton.

c. Smolensky, Paul (1996) "The initial state and Richness of the Base' in Optimality Theory," ROA 154.

d. Tessier (1996)

(54) The evolution of the child's grammar is the gradual ascent of Faithfulness constraints over Markedness, so that Junior produces ever more faithful outputs, gradually converging on fully accurate productions.

4.3 Consequences for research II

(55) Since the child's phonology seems to be "endogenous", it is worth studying as a potential "window on UG" unimpeded by accidents of history.

(56) Is child markedness the same as adult markedness?

(57) Some Phenomena Seen In Children's Outputs

a. Amahl Smith, at age 2 years, 60 days, rendered all stops as voiceless unaspirated lenis initially, voiced in medial position, and voiceless finally; thus [ˈb̥ebu] table', [aːt] hard', [ˈwəːgɪn] working'. Cf. Lac Simon, Korean, German, respectively.

b. Amahl required every consonant to be either prevocalic or final, so he produced no consonant clusters. Cf. Gokana (Hyman 1982, 1985).

c. Some children impose gaps in their stop inventories at [p] or at [g] (Ferguson (1975), Macken (1980b). Cf. Arabic, Dutch, respectively.

d. Some children voice obstruents postnasally (Ferguson 1975, 11; Locke 1983, 120, also references in Kager text). Cf. Ecuadorian Quechua (Orr 1962), Eng. dial. *Washington* ["wASIN**d**@n].

(58) Some Things Children Do that Adults Don't (with possible explanations)

a. Consonant harmony: sock = [gɑk]

(i) jaw-governed consonant articulation vs. tongue/lip-governed articulation

b. Obligatory long-distance movement of /s/: *step* = [pɛts] (Hamp 1985) (though compare historical Ilokano *taːŋis ¿ *saːŋit weep' (Aklanon taːŋis, Toba Batak taŋis), *tamis ¿ samit sweet' (Aklanon tamis, Tagalog tamis, Timugon Murut ma-tamis), and similar cases). (Blevins and Garrett 2004)

4.4 Conclusion and Propaganda

(59) Acquisition/learnability theories ideally will make progress when they are better able to connect with each other.

(60) They will do so when learnability theory becomes sufficiently concrete that it says "This is how it's done."given Linguistic theory X, Learning algorithm Y, input data Z, processing will proceed as follows and produce Grammar G."

References

Albright, Adam and Bruce Hayes. 2003. Rules vs. Analogy in English Past Tenses: A Computational/Experimental Study. *Cognition* 90:119–161.

Blevins, Juliette and Andrew Garrett. 2004. The Evolution of Metathesis. In *Phonetically-Based Phonology*, edited by Bruce Hayes, Robert Kirchner, and Donca Steriade, chapter 5. Cambridge University Press.

Friederici, Angela and Jeanine Wessels. 1993. Phonotactic knowledge of word boundaries and its use in infant speech perception. *Perception and Psychophysics* 54:287–295.

Hale, Mark and Charles Reiss. 1998. Formal and empirical arguments concerning phonological acquisition. *Linguistic Inquiry* 29:656–683.

Hayes, Bruce. 2004. Phonological acquisition in Optimality Theory: the early stages. In *Fixing Priorities: Constraints in Phonological Acquisition*, edited by Rene Kager, Joe Pater, and Wim Zonneveld. Cambridge University Press.

Hayes, Bruce and Colin Wilson. 2008. A Maximum Entropy Model of Phonotactics and Phonotactic Learning. *Linguistic Inquiry* .

Heinz, Jeffrey. 2007. The Inductive Learning of Phonotactic Patterns. Ph.D. thesis, University of California, Los Angeles.

Jusczyk, Peter, Angela Friederici, Jeanine Wessels, Vigdis Svenkerund, and Ann Marie Jusczyk. 1993. Infants' sensitivity to the sound patterns of native language words. *Journal of Memory and Language* 32:402–420.

Prince, Alan. 2002. Entailed Ranking Arguments. In *Rutgers Optimality Archive*. ROA-500, http://roa/rutgers.edu.

Prince, Alan and Bruce Tesar. 2004. Fixing priorities: constraints in phonological acquisition. In *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge: Cambridge University Cambridge: Cambride University Press.

Smith, Neilson. 1973. *The Acquisition of Phonology*. Cambridge University Press.

Smolensky, Paul. 1996. On the comprehension/production dilemma in child language. *Linguistic Inquiry* 27:720–731.

Tesar, Bruce. 1995. Computational Optimality Theory. Ph.D. thesis, University of Colorado at Boulder.

Tesar, Bruce and Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* (29):229–268.

Tesar, Bruce and Paul Smolensky. 2000. *Learnability in Optimality Theory*. MIT Press.

Tessier, Anne-Michelle. 1996. Biases and Stages in Phonological Acquisition. Ph.D. thesis, University of Massachussetts, Amherst.