# Enforcing Grammatical Restrictiveness Can Help Resolve Structural Ambiguity

Bruce Tesar
*Rutgers University*

## 1. Two Learnability Issues

Two major issues in formal language learnability are the problem of learning restrictive distributions (sometimes known as the "subset problem"), and the problem of structural ambiguity.

While substantial progress has been made in addressing each of these problems in isolation, a complication can arise when a learner is faced with a learning situation that exhibits both problems. It is possible for the two problems to interact: allowing grammars of differing restrictiveness can complicate efforts to contend with structural ambiguity.

The main result of this paper is a demonstration that a construction already proposed for learning restrictive grammars, the r-measure, can be used to contend with the complications in structural ambiguity that result from the existence of grammars of differing restrictiveness.

## 2. Learning Restrictive Distributions

### 2.1. Restrictive grammars and the r-measure

Phonotactic restrictions concern what kinds of linguistic elements can appear where. To properly learn a language, a learner must attribute such "suspicious absences" of particular elements from particular environments to the grammar. Part of the challenge in this stems from the fact that human language learners are working primarily with positive data; they do not

---

* This work was supported by NSF Award BCS-0083101. Helpful comments and discussion were provided by Alan Prince, Joe Pater, the Rutgers Optimality Reading Group, and the participants of the 2001 Workshop on Early Child Language Acquisition in Carry-le-Rouet, France. Any remaining errors are solely the responsibility of the author.

 *WCCFL 21 Proceedings*, ed. L. Mikkelsen and C. Potts, pp. 443-456. Somerville, MA: Cascadilla Press.

receive a plentiful or reliable stream of evidence directly instructing them that elements cannot appear in certain positions.

This reliance on positive data complicates learning, particularly when subset relations may exist between the languages of different grammars (Angluin, 1980, Baker, 1979). A simple illustration of subset relations can be constructed for the distribution of angma (ŋ) cross-linguistically. Angma may be banned outright from a language, as in Hawaiian. In English, angma is in the inventory, but is only permitted in coda position; it is banned from syllable onsets. Even less restrictive is Vietnamese, which permits angma in both onset and coda position (Thompson, 1987). A learner of English has positive evidence requiring that it reject the Hawaiian grammar, but does not have direct evidence requiring rejection of the Vietnamese grammar. The correctness of the English grammar must be inferred from the lack of angma in onset position.

The subset problem is particularly evident when considering early phonotactic learning. At this stage, the learner is not yet performing morphological analysis of words; each word is treated as monomorphemic. As a consequence, the learner is not in a position to identify phonological alternations, which could otherwise provide evidence for at least some phonotactic restrictions. At this early stage, the learner is trying to find a grammar that is consistent with all of the data, but the possibility of subset relations means that there will often be many such grammars. The learner must further try to find the most restrictive grammar consistent with the data; one that permits all the attested forms, but as few others as possible.

This problem is formalized here using Optimality Theory and the following assumptions. The learner is presumed to adopt, for each datum, an underlying form that is identical to the surface overt form. The goal of the learner is to find a grammar which maps each datum to itself (more precisely, to a structural description which is overtly identical to the datum), but does so for the minimum number of unattested forms.[1]

An approach to this learning problem has been developed in the form of the Biased Constraint Demotion (BCD) algorithm (Prince and Tesar, 1999, Prince and Tesar, to appear); for a similar approach see also (Hayes, to appear). The BCD approach focuses on the observation that, to an approximation, restrictiveness in Optimality Theory can be characterized by the extent to which markedness constraints dominate faithfulness constraints; more markedness constraints dominating more faithfulness constraints tends to yield a more restrictive grammar (Demuth, 1995, Gnanadesikan, 1995, Sherer, 1994, Smolensky, 1996, van Oostendorp, 1995).

1 Such a grammar is known as a *restricted identity map* (Prince and Tesar, 1999).

The BCD approach is developed from a formal characterization of what it means to have “more markedness constraints dominate more faithfulness constraints.” The formal characterization provided by Prince and Tesar is the *r-measure*. This measure assigns a number to a constraint hierarchy by adding together, for each faithfulness constraint, the number of markedness constraints that dominate it. The higher the r-measure for a given grammar, the more restrictive it is predicted to be, relative to other grammars using the same constraints. To illustrate, consider the following two constraint hierarchies:

(1) a. M1 ≫ F1 ≫ M2 ≫ M3 ≫ F2 ≫ F3

b. M1 ≫ M2 ≫ F1 ≫ F2 ≫ F3 ≫ M3

Grammar (1)a has an r-measure of 1+3+3=7, while grammar (1)b has an r-measure of 2+2+2=6. The r-measures predict that grammar (1)a should be the more restrictive.

BCD uses the r-measure as a design principle; the algorithm is designed to find the consistent hierarchy with the highest r-measure. However, it needn’t actually compute r-measures for entire constraint hierarchies in the process.

### 2.2. An illustration of BCD

The BCD algorithm is a modified version of Recursive Constraint Demotion (RCD), an earlier algorithm for ranking constraints (Tesar and Smolensky, 1994, Tesar and Smolensky, 2000). Both algorithms accept as input a set of mark-data pairs, and return a constraint hierarchy consistent with the pairs (provided one exists).[2] Both algorithms also construct the constraint hierarchy top-down, first ranking the highest constraints, then the next-highest, and so on. But while RCD places into each stratum of the hierarchy all unranked constraints eligible to be ranked, BCD postpones the placement of faithfulness constraints into the hierarchy for as long as possible, to the effect of ranking faithfulness constraints lower relative to the markedness constraints.

As an illustration, consider the set of mark-data pairs in Table 1.

2 Both algorithms also work with the use of error-driven learning to generate informative mark-data pairs from positive data. A combination of RCD and error-driven learning is sometimes labeled Multi-Recursive Constraint Demotion (MRCD).

**Table 1 The mark-data pairs before the first pass.**

| | *win ~ lose* | NOANGMAONS | NOVELAR | NOLABIAL | IDPLACE |
|---|---|---|---|---|---|
| /*iŋ*/ | *iŋ ~ in* | | L | | W |
| /*mi*/ | *mi ~ ŋi* | W | W | L | W |
| /*im*/ | *im ~ iŋ* | | W | L | W |

The pairs are shown in comparative tableau format (Prince, 2000). This illustration includes one faithfulness constraint, IDPLACE, and three markedness constraints. The BCD algorithm first identifies the constraints eligible to be ranked at the top of the hierarchy; such constraints must not prefer any losers (they should have no L marks in their columns). There are two such constraints: NOANGMAONS and IDPLACE. The RCD algorithm would have placed both constraints at the top of the hierarchy. BCD, however, notes that NOANGMAONS is a markedness constraint, while IDPLACE is a faithfulness constraint, and places NOANGMAONS only at the top of the hierarchy.

The algorithm then removes all mark-data pairs accounted for by the ranking thus far; in this example, the middle mark-data pair is removed, due to the fact that the ranked constraint, NOANGMAONS, prefers the winner for that pair. The remaining data for the algorithm are given in Table 2.

**Table 2 Remaining data before the second pass.**

| | *win ~ lose* | NOVELAR | NOLABIAL | IDPLACE |
|---|---|---|---|---|
| /*iŋ*/ | *iŋ ~ in* | L | | W |
| /*im*/ | *im ~ iŋ* | W | L | W |

BCD now performs the same steps, recursively applying to the remaining data. Of the three constraints, only one is eligible to be ranked next: IDPLACE. Even though IDPLACE is faithfulness constraint, BCD has no choice but to place it next in the constraint hierarchy. Since IDPLACE prefers the winner in all of the remaining mark-data pairs, the rest of the data have been accounted for, and the remaining constraints may be placed at the bottom of the hierarchy. The constraint hierarchy constructed by BCD for this example is shown in (2).

(2) NOANGMAONS ≫ IDPLACE ≫ {NOVELAR, NOLABIAL}

Note that the r-measure of the hierarchy returned by BCD is 1. This is higher than the r-measure of the hierarchy that would have been returned by RCD. That hierarchy, shown in (3), has an r-measure of 0.

(3) {NOANGMAONS, IDPLACE} ≫ {NOVELAR, NOLABIAL}

Like RCD, the BCD algorithm is guaranteed to find a hierarchy consistent with the data if one exists, and has a very low data complexity. By postponing the placement of faithfulness constraints into the hierarchy until required, BCD acts to return that data-consistent hierarchy with the largest r-measure. In this way, BCD seeks to return the most restrictive grammar consistent with the data.

## 3. Resolving Structural Ambiguity

Structural ambiguity results when the overt form that is directly audible to the learner can be assigned more than one full structural description. A structural description which overtly matches a particular overt form will here be called an *interpretation* of that overt form. Structural ambiguity arises in many domains; one well-studied domain is metrical stress (Dresher and Kaye, 1990, Dresher, 1999, Tesar, 1998, Tesar, 2000). Suppose a learner hears an overt form consisting of three syllables with medial main stress: σσ́σ. This form can be assigned (at least) two different interpretations: one with a left-aligned iambic foot, (σσ́)σ, and one with a right-aligned trochaic foot, σ(σ́σ).

This ambiguity complicates learning when the different interpretations are consistent with different grammars, such as grammars assigning iambic vs. trochaic feet. The challenge takes the form of an interdependence between the grammar and the interpretation of the overt forms: determining the correct grammar depends upon determining the correct interpretations for the data, and vice-versa.

When a learner encounters an truly ambiguous datum, it cannot resolve the ambiguity in isolation. Other data from the language must be brought to bear on the question. The learner needs to find a grammar which simultaneously permits some interpretation of each encountered form. That problem is here formalized in a way very similar to the restrictive distributions problem. The learner adopts, for each datum, an underlying form that is identical to the surface overt form, and the goal of the learner is to find a grammar which maps each such underlying form to a fully faithful structural description (one with an overt portion identical to the original datum).

An approach to this learning problem has been developed in the form of the Inconsistency Detection Learner (Tesar, 2000). This approach is based upon the observation that if an incorrect interpretation is assigned to a particular form, there may not exist a grammar admitting both that (incorrect) interpretation and the correct interpretations of the other forms of the language. When a set of structural descriptions has the property that no possible grammar exists admitting all of them as simultaneously grammatical, that set of structural descriptions will be labeled *inconsistent*.[3] Under present assumptions, if a learner determines that a set of interpretations of overt forms is inconsistent, the learner may safely conclude that at least one of the interpretations in that set is not the correct one for its overt form. The Inconsistency Detection Learner (IDL) attempts to resolve structural ambiguity of a form by actively entertaining the different possible interpretations of an ambiguous form, and testing each of them to see which ones are inconsistent with other data of the language; an interpretation which is inconsistent with other data already observed may be discarded.

Whenever the IDL needs to learn from an observed overt form, and that form is structurally ambiguous, it constructs all possible interpretations of that overt form, and separately tests each for consistency with any data already observed. Any combinations of interpretations that are inconsistent are discarded; others are kept. This can result in the learner at times possessing more than one set of interpretations of data that are consistent, each implicating a different grammar, with the learner waiting for further data to determine which hypothesis is correct.

For example, if a learner encounters $\sigma\acute{\sigma}\sigma$ as its first datum, it will construct the two interpretations, and determine that each is consistent with different grammars. As a consequence, the learner will hold onto both possibilities as distinct grammatical hypotheses, and wait for subsequent data to disambiguate. If the next form encountered by the learner is $\grave{\sigma}\sigma\acute{\sigma}\sigma$, and the learner determines that the only workable interpretation of this form is $(\grave{\sigma}\sigma)(\acute{\sigma}\sigma)$, then the learner will check that interpretation for consistency with each of the interpretations of the previous overt form. This results in the construction of two sets of interpretations, shown in (4).

(4) a. $\{ (\sigma\acute{\sigma})\sigma, (\grave{\sigma}\sigma)(\acute{\sigma}\sigma) \}$

b. $\{ \sigma(\acute{\sigma}\sigma), (\grave{\sigma}\sigma)(\acute{\sigma}\sigma) \}$

The learner is now able to determine that the set in (4)a is inconsistent, because there is no grammar that assigns left-aligned iambic head feet to three syllable words and right-aligned trochaic head feet to four-syllable words. It may thus be discarded. The set in (4)b, on the other hand, is consistent, and the grammar hypothesis generating the forms is kept. Thus,

3 The fact of inconsistency of any particular set of structural descriptions is, of course, completely dependent upon the linguistic theory being used, specifically upon the space of possible grammars provided by that theory.

the tri-syllabic form is disambiguated in favor of the trochaic interpretation, due to the inconsistency of the iambic interpretation with other data.

The determination of whether a set of structural descriptions is consistent can be performed by a combination of error-driven learning and RCD. This takes advantage of the fact that RCD, when posed with an inconsistent set of data, will detect that fact rapidly, informing the learner conclusively that no grammar exists that is consistent with all of the data. This is illustrated in Table 3, which depicts a set of inconsistent mark-data pairs. Note that every constraint prefers at least one loser, meaning that no matter which constraint is placed into the ranking next, the data cannot be satisfied. This gives the learner a clear indication that the data are inconsistent.

**Table 3 Inconsistent data**

| *win ~ lose* | MAIN-LEFT | MAIN-RIGHT | TROCHAIC | IAMBIC |
|---|---|---|---|---|
| (σσ́)σ ~ σ(σ́σ) | W | L | L | W |
| (σ̀σ)(σ́σ) ~ (σσ́)(σσ̀) | L | W | W | L |

Given consistent data, RCD will return a grammar generating all of the structural descriptions. Thus the same procedure serves both to determine when a combination of interpretations is inconsistent, and to select the appropriate grammar when a combination is consistent.

Previously, the IDL has been tested in simulations on data generated from OT systems for metrical stress grammars. In those cases, a sufficient amount of data would usually uniquely determine the interpretation of each overt form: there was only one assignment of interpretations to overt forms that was consistent. The IDL proved to be quite computationally efficient in those cases, determining the correct assignment of interpretations and the correct grammar with a rather modest amount of computing effort. See (Tesar, 2000) for details.

## 4. Solving Both At Once

### 4.1. The problem

In approaching linguistic systems exhibiting both grammars of differing restrictiveness and structural ambiguity, it is natural to seek to combine the proposed solutions to the two problems. It is actually quite straightforward to combine the two approaches. Previously, the IDL has been implemented using RCD as the algorithm for detecting inconsistency and determining the grammar for consistent sets of interpretations. BCD, the algorithm used for learning restrictive grammars, is a variation on RCD that preserves all of the key properties that the IDL depends upon. BCD detects inconsistency the same way RCD does, and returns a grammar for a consistent set of interpretations; it simply goes further in actively seeking the most restrictive such grammar. Thus, the two solutions may be combined by implementing the IDL with BCD. It turns out, however, that this straightforward combination is not by itself sufficient.

When there is only one consistent combination of interpretations for a set of data, inconsistency detection alone will succeed in solving the problem of structural ambiguity, and the only issue is how long it will take the learner to complete the process. If there are several consistent combinations of interpretations for the data, however, IDL may return several hypothesized grammars, corresponding to several different combinations of interpretations. Allowing grammars of differing restrictiveness can be one way to create circumstances in which all of the data generated by a particular grammar admit multiple consistent sets of interpretations.

Recall the angma distribution facts described in section 2.1. Now consider a particular overt form, *iŋi*. The critical fact about this form is the inter-vocalic angma; it is structurally ambiguous between onset and coda position. The onset interpretation, *(i)(ŋi)*, is the correct one for languages permitting angma in onset position (the parentheses are now denoting syllable boundaries). The coda interpretation, *(iŋ)(i)*, is the correct one for languages banning angma from onset position. Since both interpretations are admitted by possible grammars, the ambiguity can only be resolved by reference to other data.

Now suppose that the only other relevant overt form encountered by the learner is *iŋ*, which is unambiguously analyzed as *(iŋ)*, with angma in coda position. Following the IDL, the learner now has two sets of interpretations, {*(iŋ)*, *(iŋ)(i)*} and {*(iŋ)*, *(i)(ŋi)*}, and evaluates each for consistency. This is where the complication comes in: both sets are consistent. The first set of interpretations is consistent with a grammar banning angma from onsets but permitting them in codas; the second set is consistent with a grammar permitting angma in both onsets and codas.

Given that no data appear with angma unambiguously in onset position, the learner needs to select a grammar banning angma from onset position (the more restrictive grammar). The BCD approach to learning restrictions is not by itself sufficient to solve the problem here. BCD is designed to find the most restrictive grammar consistent with some set of interpretations. Given the first set of interpretations, {*(iŋ)*, *(iŋ)(i)*}, BCD will correctly return a grammar banning codas from onset position. But BCD itself is not designed to *choose between different sets* of interpretations.

BCD is a restrictiveness-based method for finding a grammar for a given set of interpretations. The IDL is a consistency-based method for choosing between different sets of interpretations. What is needed for the problem just described is a restrictiveness-based method for choosing between different sets of interpretations, one that will not require giving up the existing benefits of BCD and IDL. Such a method is described in the next section.

## 4.2. The proposal

The BCD algorithm is designed to find, from among the grammars consistent with a set of structural descriptions, that grammar with the highest r-measure. The r-measure figures implicitly in the design of BCD. The proposal here is that the r-measure can be more explicitly employed when comparing grammars for different sets of interpretations. Specifically, if a learner employing the IDL with BCD ends up with more than one grammar hypothesis, each consistent with the data but involving different interpretations of the data, then the learner should compute the r-measure of each hypothesis, and select the one with the highest r-measure.

It bears emphasizing that this restrictiveness-based method for choosing between different sets of interpretations is an addition to, not a replacement for, the consistency-based method used by IDL. The consistency criterion of standard IDL is actively enforced throughout learning: each time different interpretations of an overt form are constructed and combined with sets of interpretations of prior data, the resulting sets are checked for consistency, and any inconsistent sets are discarded prior to the learner's considering the next piece of data. The restrictiveness criterion of the r-measure is employed only after some appropriate amount of data have been seen,[4] and the learner has concluded that there will unlikely be any further data which are inconsistent with any of the learner's current grammar hypotheses.

It is quite important that the consistency criterion be actively enforced from the beginning, as done by IDL; otherwise, the number of distinct sets of interpretations retained by the learner would grow exponentially, requiring of the learner implausibly large amounts of memory and computational effort.

## 4.3. The learning simulations

To evaluate the effectiveness of this solution, a computer simulation of the learning algorithm was run, applying it to data generated by a linguistic system providing a basic feature-based analysis of the angma-distributions described above. There are two main questions to be answered by the

4 I will not here attempt to characterize exactly how a learner determines when enough data have been observed.

simulations: (a) will the inconsistency detection of IDL keep the number of sets of interpretations held by the learner to a reasonable number; (b) will the r-measure select the correct (most restrictive) grammar?

#### 4.3.1. The linguistic system

The linguistic system used is a combination of Basic CV Syllable Theory (Prince and Smolensky, 1993) and a simple segmental feature system. The syllable theory is like Prince and Smolensky's, but with the correspondence faithfulness model. All syllabic positions contain exactly one segment (enforced by GEN), and the sets of possible consonants and vowels are disjoint. All syllables must have nuclei (enforced by GEN). This component contributes five violable constraints: ONSET, NOCODA, MAX, DEPONS, DEPNUC.

The consonant system uses three features: place, manner, and voicing. The place feature has three values: alveolar, labial, and velar. The manner feature has two values: stop and nasal (effectively standing in for obstruent and sonorant). The voicing feature has two values: voiced and unvoiced. Each feature has an Ident faithfulness constraint, for a total of three: IDPLACE, IDMANNER, IDVOICING. There are also markedness constraints against all but one value for each feature: NOVELAR, NOLABIAL, NONASAL, NOVOICED. One inviolable asymmetry is enforced by GEN in this system: voiceless nasals are universally banned.

The simulated system also has a two-feature vowel system, involving four constraints, NOLOW, NOBACK, IDHEIGHT, and IDEXTENT. However, the vowel system does not interact in any way with the points of interest here. To simplify the data generation, all forms generated for the learning simulations used only a single vowel.

An additional constraint provides the analysis of the angma distribution: NOANGMAONSET. This constraint is violated by an angma in onset position, and directly conflicts with both ONSET and NOCODA on the syllabification of VŋV sequences.

The system has a total of 17 constraints, and the universal inventory consists of 9 consonants and 4 vowels. To put the problem in perspective, in searching for the correct grammar, the learner is searching a space of 17! = 355,687,428,096,000 constraint rankings. In learning a language in this system, the learner is simultaneously learning the segmental inventories, the syllable structure, and the syllabic segmental restrictions of the language.

#### 4.3.2. The test language

The only syllabic / segmental interaction that can be analyzed by this linguistic system involves angma, as a result of NOANGMAONSET. The interpretation of intervocalic occurrences of other consonants can be determined by inconsistency detection alone. When the learning algorithm is applied to data from languages that, for instance, ban all nasals from the inventory (including angma), it succeeds in learning the correct grammar without needing to explicitly compute r-measures; the inconsistency detection of IDL has already eliminated all but one grammar hypothesis.

The problem of interest is resident in the target language in which onsets are optional, angma is banned from onset position, and only angma is permitted in coda position. Thus, not only is intervocalic angma structurally ambiguous, but there is no evidence involving other consonants to affirm that codas are allowed.

(5) Allowed in the target language: *(ti) (i) (iŋ) (i)(ti) (iŋ)(i)*

(6) NOT allowed in the target language: *(ŋi) (it)*

The key ranking relationships that give rise to this target language are given in (7).

(7) NoAngmaOnset ≫ {Max, IDManner} ≫ NoCoda ≫ DepNuc

The data presented to the learner consisted of overt forms for the grammatical structural descriptions of the language. The crucial ambiguous overt form, *iŋi*, was ordered first, to ensure that the learner created separate hypotheses for each interpretation. The learner then processed each overt form of the data set, one at a time, until it possessed grammar hypotheses that were consistent with all of the data.

#### 4.3.3. The results

When the learner encountered the first overt form, *iŋi*, it created two hypotheses, one for the onset interpretation, and one for the coda interpretation. It then processed all of the remaining overt forms, following the IDL algorithm. At the end of overt form processing, the learner possessed two grammar hypotheses. Grammar 1, shown in (8), was based on the onset interpretation of *iŋi*, and permitted angma in both onset and coda positions; it had an r-measure of 53. Grammar 2, shown in (9), was based on the coda interpretation of *iŋi*, and banned angma from onset position; it had an r-measure of 58. The learner, on the basis of the r-measures, selected grammar 2. This was the correct choice.

(8) Grammar 1: {NoLabial, NoLow, NoBack} ≫ Max ≫ IDManner ≫ {NoNasal, NoVoice} ≫ DepOns ≫ Onset ≫ IDPlace ≫ {NoVelar, NoAngmaOnset} ≫ NoCoda ≫ {DepNuc, IDVoice, IDHeight, IDExtent}

(9) Grammar 2: {NoLabial, NoLow, NoBack, NoAngmaOnset} ≫ Max ≫ IDManner ≫ {NoNasal, NoVoice} ≫ IDPlace ≫ {NoCoda, NoVelar} ≫ DepOns ≫ Onset ≫ {DepNuc, IDVoice, IDHeight, IDExtent}

The differing restrictiveness between the two grammars is illustrated in Table 4, which shows grammatical overt forms for each. Grammar 2 is more restrictive, banning the forms with syllable-initial angma.

**Table 4 The relative overt form inventories of the two grammars.**

| | | | | | |
|---|---|---|---|---|---|
| Grammar 1 | *ti i* | *tiŋ iŋ* | *ŋi ŋiŋ* | *tiŋi iŋi* | *ŋiti* |
| Grammar 2 | *ti i* | *tiŋ iŋ* | | *tiŋi iŋi* | |

This provides answers to the two questions raised earlier. First, inconsistency detection was sufficient to restrict the learner to only two grammar hypotheses in the end. Second, the r-measure correctly identified the more restrictive of the two grammars.

## 5. Conclusions

The angma distribution analysis provides a simple example of how grammars of differing restrictiveness can complicate a learner's efforts to contend with structural ambiguity. The complication arises when an incorrect interpretation of an overt form, *(i)(ŋi)* in the example, can nevertheless be consistent with the other data of the language via a less restrictive grammar (one permitting angma in both onsets and codas) than the correct grammar (one banning angma from onsets). This complication means that, while inconsistency detection can still do a lot of work, it cannot by itself fully disambiguate all of the forms, even once all relevant positive data have been observed.

I have proposed in this paper an algorithm for learning designed to simultaneously deal with the learning of restrictive distributions and structural ambiguity. It combines solutions that have been proposed for each of the two problems in isolation, Biased Constraint Demotion and the Inconsistency Detection Learner, in a straightforward way, and adds one additional component, the explicit comparison of grammars with respect to their r-measures. The combination of BCD and IDL is achieved by simply employing BCD in the IDL algorithm where RCD had previously been used.

The comparison of grammars with respect to their r-measures does not introduce any major new theoretical machinery. The r-measure had already been proposed in the analysis of BCD as the basis for the design of the algorithm. While the r-measure already existed implicitly in the understanding of the functioning of BCD, the present proposal finds further benefit to the r-measure, explicitly computing the r-measure for a select set of grammars, and directly comparing them.

Using the r-measure to select among grammar hypotheses that result from different interpretations of ambiguous overt forms directly addresses the complication introduced by differing restrictiveness into the problem of structural ambiguity. It adds, to the IDL's existing criterion of mutual consistency, an additional criterion of relative restrictiveness. The resulting algorithm uses both criteria in simultaneously determining the correct grammar and the correct interpretations of ambiguous forms.

This work provides further support for the active enforcement of grammatical restrictiveness throughout learning. This view, previously advocated by Prince and Tesar, contrasts with the view of achieving restrictiveness by positing a restrictive initial state (typically, markedness constraints over faithfulness constraints), and relying on ranking conservatism to make only minimal, restrictiveness-preserving changes in response to data (Itô and Mester, 1999, Smolensky, 1996).

An interesting issue for future research is the extent to which restrictiveness criteria, here embodied as comparison of r-measures, can ultimately contribute to contending with structural ambiguity. In the angma distribution example investigated in this paper, the amount is structural ambiguity is relatively small when compared to cases like the stress languages previously used to test the IDL. That permitted the simple strategy of using only inconsistency detection as new overt forms were encountered, and comparing r-measures only after sufficient data had been seen. Larger language models may well exhibit both more structural ambiguity and more interaction between structural ambiguity and relative restrictiveness of grammars. A learner might make use of the r-measure comparison for rejecting significantly less restrictive grammars at several points during learning, rather than just at the end. It remains to be seen whether a more complex learning strategy of that sort can be developed which would succeed. If such a strategy exists, it would resolve the currently lingering question of when the r-measure comparison operation should be performed by a learner that is presented not with a fixed batch of data (as in the above simulations) but an on-going, non-ending supply of data (as is the case for children).

## References

Angluin, Dana. 1980. Inductive Inference of Formal Languages from Positive Data. *Information and Control* 45:117-135.

Baker, C. Lee. 1979. Syntactic theory and the projection problem. *Linguistic Inquiry* 10:533-581.

Demuth, Katherine. 1995. Markedness and the Development of Prosodic Structure. In *Proceedings of the North East Linguistics Society 25*, ed. Jill Beckman, 13-25. Amherst, MA: GLSA, University of Massachusetts.

Dresher, B. Elan, and Kaye, Jonathan. 1990. A Computational Learning Model for Metrical Phonology. *Cognition* 34:137-195.

Dresher, B. Elan. 1999. Charting the Learning Path: Cues to Parameter Setting. *Linguistic Inquiry* 30:27-67.

Gnanadesikan, Amalia. 1995. Markedness and Faithfulness Constraints in Child Phonology. Ms., Linguistics Dept., University of Massachusetts, Amherst, MA.

Hayes, Bruce. to appear. Phonological Acquisition in Optimality Theory: The Early Stages. In *Fixing Priorities: Constraints in Phonological Acquisition*, eds. René Kager, Joe Pater and Wim Zonneveld. Cambridge University Press.

Itô, Junko, and Mester, Armin. 1999. The structure of the phonological lexicon. In *The Handbook of Japanese Linguistics*, ed. Natsuko Tsujimura, 62-100. Malden, MA and Oxford, UK: Blackwell Publishers.

Prince, Alan, and Smolensky, Paul. 1993. Optimality Theory: Constraint Interaction in Generative Grammar. Ms., Rutgers University and the University of Colorado.

Prince, Alan, and Tesar, Bruce. 1999. Learning Phonotactic Distributions. Technical Report RuCCS-TR-54, Rutgers Center for Cognitive Science, Rutgers University, New Brunswick, NJ.

Prince, Alan. 2000. Comparative Tableaux. Ms., Department of Linguistics, Rutgers University, New Brunswick. ROA-376.

Prince, Alan, and Tesar, Bruce. to appear. Learning Phonotactic Distributions. In *Fixing Priorities: Constraints in Phonological Acquisition*, eds. René Kager, Joe Pater and Wim Zonneveld. Cambridge University Press.

Sherer, Tim. 1994. Prosodic phonotactics. PhD. dissertation, Linguistics, University of Massachusetts.

Smolensky, Paul. 1996. The Initial State and "Richness of the Base" in Optimality Theory. Technical Report JHU-CogSci-96-4, The Johns Hopkins University, Baltimore, MD. ROA-154.

Tesar, Bruce, and Smolensky, Paul. 1994. The Learnability of Optimality Theory. In *The Thirteenth West Coast Conference on Formal Linguistics*, eds. Raul Aranovich, William Byrne, Susanne Preuss and Martha Senturia, 122-137. CSLI.

Tesar, Bruce. 1998. An Iterative Strategy for Language Learning. *Lingua* 104:131-145.

Tesar, Bruce. 2000. Using inconsistency detection to overcome structural ambiguity in language learning. Technical Report RuCCS-TR-58, Rutgers Center for Cognitive Science, Rutgers University, New Brunswick, NJ. ROA-426.

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

Thompson, Laurence C. 1987. *A Vietnamese Reference Grammar*. Honolulu: University of Hawaii Press.

van Oostendorp, Marc. 1995. Vowel quality and phonological projection. PhD. dissertation, Linguistics, Tilburg University.