Charting the Learning Path: Cues to Parameter Setting

B. Elan Dresher
University of Toronto

Current approaches to the problem of learnability of grammars assume a highly constrained theory of Universal Grammar (UG), within which cross-language variation is kept to certain limits. These limits are set, depending on one’s theory, either by a series of variable parameters which learners must fix at their correct values (Chomsky 1981), or by a series of constraints which learners must correctly rank (Prince & Smolensky 1993). An explanatory theory ought to specify how the learner sets the parameters or ranks the constraints on the basis of relevant input data.

There are two fundamental problems we must overcome in developing a learning model. The first is that parameters and constraints interact in complex ways, and it is difficult to reliably discern what specific contribution each one makes to the whole. A learner whose hypothesized grammar does not successfully account for the target input would have no reliable information as to the nature of the error. We can call this the Credit Problem (Clark 1989 calls this the Selection Problem). A second fundamental problem is that parameters and constraints are stated in terms of abstract entities which the learner is not initially able to identify. For example, metrical theory is couched in terms of concepts such as heavy syllable, head, constituent, and projection. These entities do not come labelled as such in the input, but must themselves be constructed by the learner. Since parameters are stated in terms of metrical theory, whereas the cues to these parameters must be stated in terms of observable data, it is an empirical issue as to what the correct cue to a given parameter is (the same holds if the problem is construed as one of constraint ranking). We can call this the Epistemological Problem:

(1) Two fundamental problems

1. The Credit Problem: When there is a mismatch between a target form and a learner’s grammar, there is no way of reliably knowing which parameters/constraints must be reset to yield a correct output.

2. The Epistemological Problem: There is a gap between the vocabulary in terms of which parameters/constraints are couched and the learner’s analysis of the input.

These problems make it a challenge to devise a reliable procedure that guarantees that the learner will converge on the target grammar.

*I would like to thank Ted Gibson, Norbert Hornstein, Alana Johns, Ken Wexler, and audiences at MIT and the University of Toronto for useful comments. Needless to say, none of the above necessarily agree with the views expressed here. I am grateful for the support of SSHRC research grant 410-92-0885. This paper was presented at WECOL 1994, UCLA, and is to appear in the Proceedings of that meeting.
1. A Cue-based Learner (Dresher & Kaye 1990)

The model of Dresher & Kaye (1990), which is a learning model for a parametric version of metrical phonology, was designed as an attempt to overcome these problems in one area of phonology, though the principles are intended to hold in other domains also. I will sketch some general properties of the model, and briefly show how they work in an example case. Then I will consider some alternative approaches which have been recently proposed; I will argue that they fail to adequately address one or both of the fundamental problems.

Some of the main features of the Dresher & Kaye (1990) model are listed in (2):¹

(2) Properties of a cue-based learner (Dresher & Kaye 1990)
A. UG associates every parameter with a cue.
B. A cue is not an input sentence or form but is something that can be derived from input.
C. Cues must be appropriate to their parameters in the sense that the cue must reflect a fundamental property of the parameter, rather than being fortuitously related to it.
D. What the correct cue to any given parameter is must be empirically determined (by the linguist not the learner, to whom it is supplied by UG). There is thus no parameter-independent general algorithm for parameter setting.
E. Parameter setting proceeds in a (partial) order set by UG: this ordering reflects dependencies among cues, and specifies a learning path. The setting of a parameter later on the learning path depends on the results of earlier ones.
F. A parameter which has a default state remains in it until the learner detects its cue, which acts as the trigger to move to the marked setting. Symmetrical parameters (e.g. directional parameters) may have positive cues for both values.
G. The learning strategy is loosely speaking ‘deterministic’, in the sense of Marcus (1980) and Berwick (1985), in that the learner may not backtrack or undo parameter settings that have already been set. Some such restriction is necessary if the learner is to be prevented from getting into infinite loops.²
H. Determinism does not hold in the following case: when a parameter is set to a new value, all parameters which depend upon it (follow it in the order) revert to default.
I. Cues are local in the sense that each decision depends on finding a specific configuration in the input, and acts on this without regard to the final result. Hence, learners are not trying to match the input.
J. Cues become increasingly abstract and grammar-internal the further along the learning path they are.

By way of illustration, consider the core stress system of English, which for purposes

¹For further discussion of various aspects of this learning model see also Dresher (1991/to appear, 1992, 1994).
²See Nyberg (1991a, b), for detailed discussion of the merits and drawbacks of determinism. He argues for a limited nondeterministic learning model.
of this example we can consider to be the same as Latin. This stress pattern can be characterized as in (3); some words illustrating this pattern are shown in (4):

(3) English/Latin
Main stress falls on the penultimate syllable if it has a long vowel or is closed by a consonant; otherwise, main stress falls on the antepenultimate syllable.

(4) Some words
a. álgebra, Cánada, génesis, América
b. Váncü:ver, aró:ma, horf:zon, Mánitó:ba
c. agénda, appéndix, Hélsí:ki, Pára:cí:sus

Following standard accounts (e.g. Halle & Vergnaud 1987), the metrical patterns of sample words are derived from grid representations such as in (5):

(5) Acquired representations
\[
\begin{align*}
\text{a. } & x & b. & x & c. & x & \text{Line 2} \\
& (x) & \quad (x) & \quad (x) & \text{Line 1} \\
& x (x x) & <x> & (x x) & <x> & x (x) & <x> & \text{Line 0} \\
& L L L & L L & H L & L H & L & \text{Syllables}
\end{align*}
\]
American Manito:ba agenda

In these grids, \( H \) represents a heavy syllable (a syllable containing a long vowel or closed by a consonant), \( L \) a light syllable (a syllable containing a short vowel). The relative stress of a syllable is indicated by the height of its grid column. Parentheses indicate constituent boundaries. Angle brackets indicate an extrametrical syllable. In each line 0 constituent, one and only one element projects a mark on line 1: this element is the head of the line 0 constituent. Line 1 marks are similarly gathered into a constituent whose head is on line 2.

Let us assume that the grids in (5), constructed in accordance with parameters which we will take up as we proceed, are what learners of English have to arrive at. I assume also that the input that the learners have to work with consists of words associated with primitive grids which represent only the observed stress contours of each word. For the words in (5), the input (i.e. the learner's representation of the surface form) would look like (6):

(6) Initial representations
\[
\begin{align*}
\text{a. } & x & b. & x & c. & x & \text{Line 2} \\
& x & \quad x & \quad x & \text{Line 1} \\
& x x x x & x x x x & x x x & \text{Line 0} \\
& S S S S & S S S S & S S S & \text{Syllables}
\end{align*}
\]

America Manito:ba agenda

The input grids indicate the shape of the stress contour of a word, but they lack constituent boundaries and extrametricality markings: these must be supplied by the learners. Also, since the distinction between heavy and light syllables is not self-evident to begin with, \( L \) and \( H \) are replaced by \( S \), which represents any syllable.

In English, the location of stress depends on the distribution of heavy syllables, as
well as location in the word. Hence, a learner can make no progress in acquiring the correct
pattern without first determining that English distinguishes light from heavy syllables; i.e.
English stress is quantity sensitive, henceforth QS. Stress systems which do not distinguish
between syllable types are called quantity insensitive, or QI. The task, then, is to discover
that English stress is QS without making use of the generalization in (3), since this pattern
cannot itself be discerned until one distinguishes between light and heavy syllables.

One operation that is available to a learner at this early stage in the acquisition of the
system is classification. It is reasonable to suppose that learners begin with simple
representations and must be driven to adopt more complex ones. Thus, we may suppose that
the default is to assume that all syllables are the same for purposes of stress, i.e. assume that
stress is QI. Because all syllables have the same status in QI systems, it follows that words
with the same number of syllables are all alike from the point of view of the metrical
parameters. In QS systems, by contrast, this is not the case, as is demonstrated by the
equivalence classes of word types shown in (7):

(7) Word classes in QI and QS systems

<table>
<thead>
<tr>
<th>QI: Syllable = S</th>
<th>QS: Syllable = H or L</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 syllable words: {SS}</td>
<td>{LL} {HL} {LH} {HH}</td>
</tr>
<tr>
<td>3 syllable words: {SSS}</td>
<td>{LLL} {HLL} {LHL} {HHL}</td>
</tr>
<tr>
<td>4 syllable words: {SSSS}</td>
<td>{LLLH} {HLLL} {LHLL}...</td>
</tr>
</tbody>
</table>

In QI systems, all words with \( n \) syllables should have the same stress contour, since
they are all effectively equivalent. Taking QI to be the default case, a learner will continue to
assume QI until it encounters evidence that words of equal length can have different stress
counts:

(8) QS

a. Subset: QI languages are subset of QS languages.
b. Default: Assume all syllables have the same status (QI).
c. Cue: Words of \( n \) syllables, conflicting stress contours (QS).

Such evidence is abundant in English, as is apparent in (4); for example, the three-syllable
words in (4a) have initial stress, conflicting with the three-syllable words in (b) and (c) which
have stress on the middle syllable; similarly, \textit{América} conflicts with \textit{Mánitoba}, and so on.
The existence of conflicting stress contours on a wide scale would lead the learner to abandon
the default hypothesis. Note that QS is not the only cause of such conflicts: the language in
question may have lexical accent, for example. A fuller specification of the learning path
would have to include means for distinguishing between QS and lexical accent, but we cannot
consider all the possibilities here (see Dresher 1994 for some discussion). Similar
considerations hold all along the line. Assuming though, that other possibilities are ruled out,
the learner is led to revise the input representations, now distinguishing between light and
heavy syllables.

Here, too, there are choices to make, because not every language has the same
characterization of what a heavy syllable is. Some languages do not count closed syllables
with short vowels as heavy. (9) gives a slightly oversimplified picture of the possibilities, but
one we will adopt here: we will assume that syllables that end with a short vowel (short open syllables) are universally light, and that syllables with long vowels are universally heavy. Closed syllables may go either way:

(9) Light and heavy syllables
    \[ \begin{array}{ccc}
    \text{Always Light (L)} & \text{L or H} & \text{Always Heavy (H)} \\
    \ldots V. & \ldots V.C. & \ldots V.V \\
    \end{array} \]

In order to determine which style of QS English adopts, we can continue with the classification test we used to diagnose QS in the first place. We assume that when learners determine that a language is QS, they revise their initial representations, now characterizing syllables as being either L or H. Suppose that the initial revision incorrectly assumes that closed syllables are light; we would arrive at the word classes in (10):

(10) Assuming QS, closed syllables light: conflicting words
    L L L: ál.ge.bra (lx) a.gén.da (x/x) Hèl.sín.ki (vx)
    L L L L: A.mé.ri.ca (x/xx) Pà.ra.cé.l.sus (v/x)

The new representations still contain conflicting words: thus, words of the pattern LLL do not all have the same stress contour, nor do words of the pattern LLLL. These conflicts, which would again exist on a large scale in the language, would serve as a trigger to try the other possibility in (9), which leads to representations in which closed syllables count as heavy:

(11) Assuming QS, closed syllables heavy: no conflicting words
    H H H: Vâncóu:ver H H L: Hèlsín.ki
    H L L: álgebra L L H: génesis
    L L L: Cánada L H L: aró:ma, agén.da
    L H H: horíf:zon, appéndix L L L L: América
    L L H L: Mánito:ba L H H H: Pàräcél.sus

These representations contain no conflicts, an indication that the representations can serve as a basis for proceeding to set further metrical parameters.

Having found the heavy syllables, what we know about the sample words in (6) is given in (12):

(12) New representations with light and heavy syllables
    a. \( \begin{array}{ccc}
    x & x & c. x \\
    \end{array} \) Line 2
    x x x x x x x
    L L L L L L H H L H L H
    America Manito:ba agenda

    Main stress is assigned by promoting either the leftmost or rightmost line 1 mark onto line 2. So, although main stress is not confined to the first or last syllable, it is limited to the first or last line 1 mark, which is the head of the first or last line 0 constituent. This fact suggests a cue for main stress, given in (13):
(13) Main stress
   a. Parameter: Project the \{left/right\}-most element of the line 1 constituent.
   b. Cue: Scan a constituent-sized window at the edge of a word. Main stress should consistently appear in either the left or right window.

It follows from (13) that we do not need to know exactly what the constituents of a word are in order to determine whether main stress is on the left or the right, but we do need to know how big a metrical constituent is. In particular, we need to know if line 0 constituents are bounded or not; for purposes of this discussion, let us limit bounded constituents to binary ones:

(14) Bounded constituent construction
   a. Parameter: Line 0 constituents are bounded.
   b. Cue: The presence of a stressed non-edge \(L\) indicates bounded constituents.

If a language has bounded constituents, then a constituent-sized window would not be more than two syllables long. By contrast, if a language does not utilize bounded constituents, the only constituents it will have, if it has any, are those created by heavy syllables and by edge rules. English has bounded constituents; how might a learner determine this? A number of possible cues come to mind, for example the presence of alternating stress, but this turns out to be a slippery cue, for various reasons. The essential difference between languages with bounded constituents and languages without them is that in a language with no bounded constituents, constituent edges must be associated either with heavy syllables, or with the edge of a word. Therefore, the only light syllable that can be stressed is one that is at a word edge. It follows that the presence of a stressed light syllable that is not at a word edge is evidence for bounded constituents.\(^3\) We adopt this as the correct cue for boundedness, given in (14b).

English has such internal stressed light syllables: an example - actually, the only example in our data set - is the word \textit{America}. Without this word, the forms in (4) would be equally analyzable as an unbounded stress system with the pattern: stress the last heavy syllable which does not occur in the final syllable; otherwise, stress the initial syllable.

We will not look at the remaining parameters here; continuing in this fashion, we can go on to specify the entire learning path for acquiring the metrical system of this language. The way this learning model addresses the Credit Problem and the Epistemological Problem should by now be clear. The Credit Problem is solved for the learner by associating each parameter with a cue: the learner always knows what to look for to set a parameter. Moreover, the learner is never asked to apportion credit for an entire form to a set of parameters. The Epistemological Problem is solved by ordering the parameters; the parameters we have discussed are ordered as in (15):

\(^3\)We abstract away here from the effects of extrametricality, which can potentially change the location of the effective edge; for further discussion, see Dresher & Kaye (1990) and Dresher (1991/\textit{to appear}).
(15) Order in which parameters must be set
   a. Syllable Quantity: Establish whether feet are QI (default) or QS.
   b. Foot size: If QI, only bounded feet are available; if QS, unbounded
      is default.
   c. Main stress: Depends on correct setting of (a) and (b).

This ordering allows for a general progression, both in the representations and in the cues, from relatively simple to more complex and more abstract. The cue for quantity sensitivity, for example, coming near the beginning of this learning path, is couched in terms that presuppose little knowledge of any details of the grammar. The learner needs only to be able to keep track of stress contours and syllables. By contrast, the cue for main stress is considerably more sophisticated in what it assumes about the grammar. If parameters were unordered, then the cues would not be able to be stated in this progressive fashion.

I would like to turn now to consider some other learning algorithms that have been proposed in the recent literature. I think that all of them represent interesting proposals; but each of them makes some crucially wrong assumption about the nature of the learning problem.

2. The Triggering Learning Algorithm (Gibson & Wexler 1994)

Let's consider first the model sketched in Gibson & Wexler (1994). Gibson & Wexler formulate a general scheme they call the Triggering Learning Algorithm (TLA):

(16) The Triggering Learning Algorithm (Gibson & Wexler 1994)

   Given an initial set of values for \( n \) binary-valued parameters, the learner attempts to syntactically analyze an incoming sentence \( S \). If \( S \) can be successfully analyzed, then the learner's hypothesis regarding the target grammar is left unchanged. If, however, the learner cannot analyze \( S \), then the learner uniformly selects a parameter \( P \) (with probability \( 1/n \) for each parameter), changes the value associated with \( P \), and tries to reprocess \( S \) using the new parameter value. If analysis is now possible, then the parameter value change is adopted. Otherwise, the original parameter value is retained.

This algorithm incorporates two constraints which are due to Robin Clark, though he does not accept them as being valid:

(17) The Single Value Constraint

   Assume that the sequence \( \{h_0, h_1, \ldots, h_n\} \) is the successive series of hypotheses proposed by the learner, where \( h_0 \) is the initial hypothesis and \( h_n \) is the target grammar. Then \( h_i \) differs from \( h_{i-1} \) by the value of at most one parameter for \( i > 0 \).

---

4See also Frank & Kapur (1993) and Niyogi & Berwick (1993) for refinements and further investigation.
The Greediness Constraint
Upon encountering an input sentence that cannot be analyzed with the current parameter settings (i.e., is ungrammatical), the language learner will adopt a new set of parameter settings only if they allow the unanalyzable input to be syntactically analyzed.

The notion of trigger is implicit in the TLA. Gibson & Wexler define triggers as in (19). Only local triggers (19b) are of real interest to us. Put informally, a local trigger is a sentence of the target language which requires the learner at a particular space to set one parameter to its correct value:

(19) Triggers (Gibson & Wexler 1994)
a. A global trigger for value \( v \) of parameter \( P_i \), \( P_i(v) \), is a sentence \( S \) from the target grammar \( L \) such that \( S \) is grammatical if and only if the value for \( P_i \) is \( v \), no matter what the values for parameters other than \( P_i \) are.
b. Given values for all parameters but one, parameter \( P_i \), a local trigger for value \( v \) of parameter \( P_i \), \( P_i(v) \), is a sentence \( S \) from the target grammar \( L \) such that \( S \) is grammatical if and only if the value for \( P_i \) is \( v \).

An example of how this learning algorithm is supposed to work is given in (20), where each square represents a setting of two syntactic parameters. The first parameter determines whether the head of Spec \( X' \) is initial (value 1) or final (0). In this case, the head is the verb (V) and its specifier is the subject (S). The second parameter similarly encodes whether the head of a complement is initial or final, here exemplified by the relation between a verb and its object (O). These two parameters define a space with four states:

(20) Parameter space: (Spec-Head f/i, Comp-Head f/i): final=0, initial=1

```
Source :---> 0,0 0,1 1,0 1,1
S V S V
S O V
V S
O V S
V S
S V O
```

Assume now that the target language is VOS (1,1), and the learner's current hypothesis is SOV (0,0). Suppose the learner hears a sentence of the form \( V O S \). This sentence is not parseable by the learner, who now determines that the current state is not correct. Even though there is only one setting of parameters that corresponds to \( V O S \), we can see that it would take a change of both parameters for the learner to reach it. This is not allowed by the Single Value Constraint, which makes available only the two neighbouring spaces. Neither space yields the target \( V O S \). Therefore, according to the Greediness Constraint, the learner cannot move. Thus, the sentence \( V O S \) is not a trigger to a learner at
Fortunately in this case, there is another type of sentence from the target that the learner will eventually hear, namely V S. V S is a trigger to a learner at (0,0), since there is a neighbouring space which parses it, namely (1,0). So the learner moves to there. From there, a further presentation of V O S, which is a trigger to a learner at (1,0), will take the learner to the target.

Gibson & Wexler point out that the TLA will not be successful in the case of subset parameters, i.e. parameters where the sentences generated by one value are a proper subset of the sentences generated under the other value; in that case, the learner who is mistakenly in the superset state will have no triggers, since all input sentences can be analyzed. They restrict their discussion to nonsubset parameters.

The main point of their paper is that the TLA does not guarantee that a learner will converge on the target, because there are nonsubset parameter sets where there are no triggers. The type of example they illustrate involves local maxima, which are triggerless islands in the parameter space.

Their illustration requires us to add one more parameter, the parameter that is responsible for verb-second effects (assuming this is one parameter). This parameter has the value 0 if the grammar is not V2, and 1 if it is. V2 has the potential to obscure the effects of the other parameters by requiring movement of the verb into second position, and some other constituent into first position. The parameter space can be diagrammed as in (21):

\begin{equation}
(21) \text{Parameter space adding V2: } 0 = -V2, 1 = +V2
\end{equation}

Suppose the target is (0,1,0): SVO with no V2. Such a language has structures as in (22a):
(22) Sample structures: target (0,1,0), source (1,1,1) is local maximum


Suppose also that the learner is currently at (1,1,1): VOS +V2, with forms as in (22b). There are some sentences that look the same in both, even though their structures are different, e.g. S V O. So learners in (1,1,1) will not move when they hear any of these. It turns out that all the potential triggers are not in spaces accessible to the learner. For example, the target string Adv S V is not parsable by the learner; but none of the three moves it can make results in this string. It would have to change two parameters to see any improvement. Therefore, the learner is stuck at a local maximum.

Local maxima, therefore, are a second threat to learners adopting the TLA. Another, mentioned theoretically by Frank & Kapur (1993) which I will illustrate with real parameters, is what we can call thrashing: the possibility that a learner can go back and forth between two or more states indefinitely. To illustrate this, we will look at the interaction of parameters of metrical theory.

To keep the problem manageable, let us assume for now that all parameters are fixed except for three. For concreteness, let's assume that main stress is on the right, feet are binary, and the rightmost syllable is extrametrical (so far, as in English nouns). The free parameters in the diagram are as follows: the first number is the value of the foot head parameter, which is 0 if set to Left, i.e. trochee, and 1 if set to Right, or iamb - in the diagram, the four boxes in the top half are trochees, the bottom four are iambs; the second parameter codes direction of construction of feet, either left to right = 0 on the left side of the diagram, or right to left = 1 on the right side; and the third number represents syllable quantity, either QI = 0 for the four inside boxes, or QS = 1 for the four outside ones:
(23) Parameter space: (Foot head, Direction, QI/QS)
Assume we keep fixed: main stress on the right, feet are binary, the rightmost syllable is extrametrical. Parameters in the diagram:
  a. Foot head: Left (Trochee) = 0  Right (Iamb) = 1
  b. Direction: Left to right = 0  Right to left = 1
  c. QI/QS:  QI = 0  QS = 1

(24) Sample forms: a = stressed on 1st syllable, b = stressed on 2nd, etc.

<table>
<thead>
<tr>
<th>Target</th>
<th>(0,1,1)</th>
<th>(0,0,1)</th>
<th>(1,0,1)</th>
<th>(1,0,0)</th>
<th>(0,0,0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. álgebra</td>
<td>a</td>
<td>álgebra</td>
<td>a</td>
<td>b</td>
<td>a</td>
</tr>
<tr>
<td>2. agenda</td>
<td>b</td>
<td>agenda</td>
<td>b</td>
<td>b</td>
<td>b *a</td>
</tr>
<tr>
<td>3. Cánada</td>
<td>a</td>
<td>Cánada</td>
<td>a</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>4. América</td>
<td>*a</td>
<td>*Ámerica</td>
<td>b</td>
<td>b</td>
<td>b *a</td>
</tr>
</tbody>
</table>

Suppose the target is (T, R, QS), i.e. (0,1,1) (as in English nouns). Four sample words are listed in (24). The correct stress patterns corresponding to the target are as in English, with stress on the first syllable in the first and third words, álgebra and Cánada, and on the second syllable in the second and fourth words, agenda and América. The notation 2b means that the second word is stressed on the second syllable; 4c means the fourth word is stressed on the third syllable, and so on. Next to each box are listed the forms generated by those parameter settings; asterisks indicate forms that are ungrammatical relative to the target. Forms in bold along the lines associated with arrows are words that could move the learner in the indicated direction.

Suppose that the learner is at (0,0,0). Of the four sample words, the learner has two correct, and differs in the second and fourth words: the learner’s grammar generates and parses agenda and América, the target has agenda and América. Neither of these words is a local trigger, according to the definition, because they do not force the learner to change one
parameter to its correct value. Suppose the learner hears the word 2b (agéndad). There are two possible moves that will result in matching this form: one is to (0,0,1), which results in a correct change of the third parameter; but the learner could also successfully account for 2b (in appearance, if not in actuality) by moving away from the target to (1,0,0), losing the correct value of the second parameter. At (1,0,0), the learner again has two out of four words correct - this time the other two words. It can resolve them by moving back to (0,0,0), a return trip which can be taken many times. This situation arises in a number of cases here, and the more types of words we add, the worse the problem will be. If there are also some built-in preferences - i.e. if given a choice the learner will prefer certain routes - it is possible for the learner to prefer the thrashing paths, and wander the parameter space indefinitely.

Gibson & Wexler consider a number of ways of overcoming the problem of local maxima. They observe that local maxima arise when the learner mistakenly gets into a +V2 state, and that the problem would not arise if the learner could be prevented from trying +V2 until it has tried -V2 options. The solution they appear to favour is to adopt default states for parameters together with requiring that parameters be set in a partial order. Thus, their model becomes closer to ours in these respects. However, they still wish to preserve the essential features of the TLA. But over and above the technical difficulties, I think the TLA runs into some serious conceptual problems which I would now like to discuss.

The essential difference between the TLA and the cue-based learner has to do with the conception of what the learner is trying to do, and what constitutes a trigger, or cue. Under the TLA, the learner is trying to match the target input forms; hence, a trigger is an actual input form. A cue-based parameter learner, by contrast, is not trying to match the target forms, but uses them as sources of cues. Thus, whereas triggers in the TLA are extensional entities, actual forms that are part of E-language, cues are intentional entities. Similarly, the two learning models treat parameter dependencies in different ways. In the cue-based learner, parameter dependencies are fixed by UG, and reflect essential properties of the parameters themselves; in the TLA, dependencies between parameters arise purely as a result of accidental features of the input.

Further, Gibson & Wexler’s account is predicated on the assumption that the target sentences come in the form of strings like those in (22), which have the form S O V, Adv Aux S O V, etc. Of course, the real target sentences that the learner sees are not in that form, but are actual utterances: John kicked the ball, Je le vois, etc. A successful analysis of the complete sentence involves not just its syntactic word order, but everything else as well: phonology, morphology, etc. So the set of parameters in play are not just those affecting word order, but all of them. Now, chances are that a learner, especially at an early stage, is unable to match even simple sentences with respect to any component of the grammar: not just word order may be off, but also morphology, inflection, segmental phonology, metrical and prosodic properties, and so on. So if a learner hears a sentence of the form S V O and is currently at SOV, a change to SVO will still not result in a complete match of the whole sentence. Similarly, any change in another type of parameter - say, a morphological parameter - might result in a successful match there, but will not be considered a success by the learner, because the word order is still not right. Recall that a learner does not know what effect any given parameter has, and is not satisfied with improvements that fall short of success. So, taken literally, the TLA would not let a learner get off the ground. This is because it requires a chain of complete successes. In any one domain, such a chain could be compiled, perhaps, by starting with small targets which can be matched, and working up
from there. But over the grammar as a whole no target is small enough to be perfectly matched, especially at early stages.

Let us suppose, then, that Gibson & Wexler intend that the learner can separate out the word order properties of a sentence from its other properties. Let’s say that success must be total only within this domain. The problem with this is that the domain of facts influencing the setting of word order parameters is not limited to word order. Suppose that pronouns can be clitics, or not. So *Je le vois* could be an example of $SOV$ (if the subject and object are not clitics), or $SV$ (if the object is a clitic, so that there is no lexical material in the actual object position), or just $V$ (if both subject and object are clitics):

(25) Representations of *Je le vois*
   
   a. Subject nonclitic, object nonclitic: $SOV$
   
   b. Subject nonclitic, object clitic: $SV$
   
   c. Subject clitic, object clitic: $V$

The learner’s analysis depends on the current state of its grammar. The terms $S$, $V$, $O$ are not primitives coming from the target, but are assigned by the learner, based on knowledge of the grammar. So we cannot limit the parameter space relevant to word order only to word order parameters. For example, if the learner is currently assuming SVO plus (25a) and hears the sentence *Je le vois*, it perceives the sentence as $SOV$. Now the learner can change word order and move to SOV plus (25a); or, without changing word order, it can move to SVO plus (25b). Clearly, word order parameters cannot be correctly set without taking into account clitic status and other such matters. But how does the learner know which group of parameters forms a subspace within which matching must be perfect? It appears that, even on Gibson & Wexler’s own account, the learner must have some idea about what sort of thing a parameter does.


I would now like to look briefly at another approach to parameter setting developed by Clark (1990, 1992), and applied to V2 changes in the history of French by Clark & Roberts (1993).

Clark believes, as we do, that it is impossible to figure out which parameters are correct and which are incorrect when the learner’s grammar does not give the right results. Unlike us, he does not believe it is possible to associate reliable cues to parameters. Rather, he believes that it is possible to assign a fitness measure which gives the relative fitness of a grammar compared to others. His idea is that parameter setting proceeds by way of a genetic algorithm which enacts a Darwinian competition of survival of the fittest. He proposes that a learner simultaneously considers a number of competing hypotheses. Initially, these hypotheses may be selected randomly. Each candidate is exposed to input which it attempts to parse. At the end of a round of parsing, the learner assesses how well each candidate did. The candidates are ranked according to their relative fitness. The fittest go on to reproduce candidates in the next generation, the least fit die out. Through successive iterations of this

---

3 A different approach motivated in part by the Credit Problem is taken by Kapur (to appear), but limitations of space preclude us from discussing it here.
procedure, the candidate set presumably becomes increasingly fit, and converges toward the correct grammar.

This approach is at the opposite pole from the cue-based learner. The cue-based learner knows why it set a particular parameter to a particular value — because it saw or failed to see a cue — but it has no way to evaluate the overall success of its grammar. The learner following the genetic algorithm has no idea what contribution any particular parameter makes, but has an exquisite sense of the overall relative success of the grammar.

The proposed fitness measure is given in (26):

\[
(\sum_{j=1}^{n} v_j + b\sum_{j=1}^{n} s_j + c\sum_{j=1}^{n} e_j) - (v_i + bs_i + ce_i)
\]

\[
(n - 1)\left(\sum_{j=1}^{n} v_j + b\sum_{j=1}^{n} s_j + c\sum_{j=1}^{n} e_j\right)
\]

where

\(v_i\) = the number of violations signaled by the parser associated with a given parameter setting;

\(s_i\) = the number of superset settings in the counter; \(b\) is a constant superset penalty < 1;

\(e_i\) = the measure of elegance (= number of nodes) of counter \(i\); \(c < 1\) is a scaling factor.

There are three main terms in the metric. The first term, \(v\), refers to the number of violations signaled by the parser associated with a given parameter setting. To the extent that a candidate parameter counter is wrong, there will be some sentences that it will fail to parse. Whereas in the Gibson & Wexler scheme the learner is told only if a hypothesis succeeds or fails, Clark proposes to quantify the failure in terms of the number of violations incurred. The sum term totals up all the violations created by all the candidates. Let’s say there are five candidates who together total 50 violations. We then subtract from the total the number of violations incurred by any candidate \(i\), and divide by the total (multiplied by \(n-1\)), and we have a measure of how well candidate \(i\) is doing compared to the rest. For example, if the candidate creates 10 violations, its score is 50 - 10 = 40 divided by some number; if the second candidate creates 30 violations, its score is 50 - 30 = 20 divided by that number, a lower score.

This term is the main component of the fitness metric. Clark builds in two other terms, scaled down by constant factors to make sure they are small relative to the \(v\) term. The second term is a superset penalty, designed to have the effect of the Subset Condition. If two candidates differ only in one subset parameter, and the target language is the subset language, they ought to score identically with respect to violations, since anything that the subset parameter value can parse the superset value can do, too. To keep the learner out of the superset, Clark builds in a penalty, the term \(s\). So if two candidates both have 10 violations, they will have equal scores of 10 (roughly, forgetting about the subtraction and division). If candidate 1 has one superset parameter value, its score will be lowered by the constant term \(b\). Candidate 2, let’s say with 2 supersets, is penalized by \(2b\). Clark (1990) suggests that \(b\) is very small, around 0.00002: it has to be much smaller than 1, since it should not count nearly as much as a violation. Whatever the number, it is enough to put candidate 1 ahead of its superset competitor. The third term, \(e\), is another refinement, a measure of elegance,
which Clark roughly equates with the number of nodes that a candidate hypothesis needs to parse the target sentences. This is to give the effect of economy, preferring simple grammars to more complex ones. Clark & Roberts argue (p. 342) that the empirical facts of French show that the constant $c$ is greater than $b$, i.e. elegance counts more than subsetness.

I would like to raise some questions about the feasibility and plausibility of the fitness metric; lacking calculations and detailed proof, these remarks have to remain at a general level. Consider, for example, the subset penalty. This penalty refers to E-language (extensional) subsets, actual subsets calculated over sentences. Clark suggests that superset parameters are listed in a table, i.e. supplied to the learner by UG. In the cue-based learner, I-language subsets are a function of the learner’s built-in learning path.

The Subset Principle, as formulated by Berwick (1985), is given in (27):

(27) Subset Principle (Berwick 1985)
Choose the subset language as the default parameter setting.

A standard example is a simplified version of the Pro-Drop (or Null Subject) parameter, illustrated in (28). A language which does not allow Pro-Drop (say English) requires that all sentences have a lexical subject; in a language which allows Pro-Drop (say Italian), sentences may appear without overt subjects. If Pro-Drop is limited to just these facts, then we observe that the set of sentences where we can generate with no Pro-Drop is a subset of the set of sentences we can generate with Pro-Drop:

(28) Pro-Drop Parameter (simplified)

```
NO --- -- John walks
(English) 
walks

>---YES
(Italian)
```

From examples like these it is easy to suppose that the subset relation is an E-language relation that applies to extensional languages, but we have argued that this is not the right way to look at it: relevant subsets are defined with respect to cues. This point can be simply demonstrated by considering again the metrical parameter which determines whether stress in a language is sensitive to quantity (QS) or not (QI). Now let us consider the relation between QS and QI systems (29). If we look only at the output forms, there is no subset relation between them: a QI system generates one set of stressed words, while a QS system generates another, perhaps overlapping; set:

(29) Quantity sensitivity does not involve extensional subsets
a. Some English words, QS: álgebra, agénda, Mánito:ba
b. If English were QI, cet. par.: álgebra, ágenda, Maníto:ba

From the point of view of a learner, however, there may be a subset relation between

---

Footnote 6: For an overview of the complexities of this parameter, see the articles collected in Jaeggli & Safir (1989).
the two values. Recall that the diagnostic we used for setting this parameter, in (8), treats QI as a subset of QS, because the number of partitions of lexical classes in QI is a subset of those in QS. A learner who starts by assuming QS in this system will not recognize that the language it is learning is really QI.

In Dresher & Kaye (1990), we show how this subset relation would be reversed if one were to adopt a different cue for this parameter. The typical distribution of syllable types in (30) suggests the cue in (31):

(30) Syllable Types in QI and QS Systems

<table>
<thead>
<tr>
<th></th>
<th>QI Systems</th>
<th>QS Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stressed</td>
<td>Heavy, Light</td>
<td>Heavy, Light</td>
</tr>
<tr>
<td>Unstressed</td>
<td>Heavy, Light</td>
<td>- , Light</td>
</tr>
</tbody>
</table>

(31) QS: alternative cue, different subset relation
a. Subset: QS languages are subset of QI languages.
b. Default: Assume QS.
c. Cue: Look for an unstressed heavy syllable.

This approach to determining quantity sensitivity is not particularly good, for it is easily fooled. Nevertheless, the example illustrates that subset relations need not be construed in terms of extensional languages. In a cue-based learner, the notion is intensional - the cues determine the subset relation. But the learner following the genetic algorithm has no idea about what any individual parameter does, and yet does know which parameters create extensional supersets. Why such an extensional relation should be part of UG needs to be explained.

It is also not clear whether a useful fitness metric can be devised for every aspect of the grammar. Consider the metrical parameters, for example. A look back at (23) shows that there is no clear correlation between the number of words correct and the distance from the target. And there are many much more dramatic examples. Imagine a language with simple alternating stress. If we change the foot parameter from trochee to iamb, every syllable will receive the wrong stress. If we then move further from the target by changing other parameter values in the wrong direction, our performance - in terms of syllables or words correct - will appear to improve. In general, depending on the situation, small changes can have big effects and big changes can have small effects. It remains to be shown that the fitness metric can provide a useful guide to a learner in these circumstances.

4. Recursive Constraint Demotion Algorithm (Tesar & Smolensky 1993)

Finally, I would like to consider the Recursive Constraint Demotion (RCD) learning algorithm proposed by Tesar & Smolensky (1993) for learning how to rank constraints in Optimality Theory. They characterize the learning problem as in (32):

(32) The learning problem (Tesar & Smolensky 1993)
The initial data for the learning problem are pairs consisting of an input and its well-formed (optimal) parse.
By input, they mean an underlying form known to the learner, not input from the target language which the learner is trying to match. They give an example of a learner learning a language which allows only CV syllables. They assume a number of universal constraints on syllable structure, some of which are given in (33). These constraints may have language-particular rankings: lower-ranking constraints may be violated to preserve higher-ranking ones:

(33) Some CV syllable structure constraints
a. ONS Syllables have onsets
b. -COD Syllables do not have codas
c. PARSE Underlying material is parsed into syllable structure

They write (p. 8), "For example, the learner...might have as an initial datum the input /VCVC/ together with its well-formed parse .□V.CV. <C>... Together with this single piece of explicit positive evidence comes a large mass of implicit negative evidence. Every alternative parse of this input is known to be ill-formed;' for example, the parse *.□V.CVC. is ill-formed. In (34), □ designates an epenthetic segment; < > designates a (deleted) segment with no phonetic representation; α < β indicates that form α is less harmonic than β; C₁ >> C₂ indicates that constraint C₁ dominates C₂:

(34) Example: syllable structure
a. Underlying form: /VCVC/ (e.g. /amuk/)
b. (Optimal) surface parse: .□V.CV. <C> (e.g. [tamul])
c. Alternative parse *.□V.CVC. (e.g. *[tamuk])
d. Conclude: .□V.CV. < .□V.CV. <C>
e. Hence: -COD >> PARSE

Thus, with respect to the unknown constraint hierarchy, the learner knows (34d) that (34b) is better than (34c). From this, the learner can conclude (34e) that -COD dominates PARSE in this language. The RCD algorithm then demotes PARSE relative to -COD. Recursive applications of this algorithm, the details of which we need not go into here, rank all the relevant constraints.

To rephrase Tesar & Smolensky's statement of the problem in other words, they are assuming that before the learner has any idea how to rank the constraints, it knows that a word whose phonetic representation is, say, [tamu] has a certain surface syllable structure as well as an underlying representation, say, /amuk/. If indeed the learner already knows this, then it is true that it can deduce that the constraints are ranked as they are. As to how the learner acquires underlying representations, this is a problem for everyone, and I do not question this assumption here. However, Tesar & Smolensky do not explain how it is that the learner can know what the well-formed surface representation is before having ranked the constraints.

In the example given, the surface parse could appear to be fairly transparent. However, we have seen that representations, even surface representations, are not fixed from the outset, but are gradually developed as the learner acquires more of the grammar. This is one way to solve the Epistemological Problem. With respect to syllable structure, there are many cases where the correct surface parse is not obvious, if we allow some segments to
sometimes appear in the nucleus and sometimes in the coda, or sometimes in a coda and sometimes in an appendix, and so on. But the Epistemological Problem in Tesar & Smolensky’s algorithm can be seen in its full force when we turn to an example from metrical theory.

Imagine that the learner encounters the word *agenda* before knowing how any constraints are ranked. The learner must assign a surface parse to this form; however, any of the parses in (35) may be possible:

(35) Some possible metrical parses of *agenda*, metrical system unknown

\[
\begin{align*}
\text{a.} & \quad x & \text{b.} & \quad x & \text{c.} & \quad x & \text{d.} & \quad x & \text{e.} & \quad x \\
& x & & x & & x & & x & & x \\
& x & x & x & & x & x & x & & x & x \\
\end{align*}
\]

The correct parse, by Tesar & Smolensky’s assumption, is already known before any constraints have been ranked - but, assuming the parsing is not given directly by the acoustic signal, how can this be?

Suppose we drop the assumption that the surface representation is known beforehand: how would the learning algorithm go? The learner can’t rank any constraints because it doesn’t know which candidate wins.

I will not attempt to solve this problem here, but let’s consider what kinds of solutions there may be:

1. The surface parse may be given directly in the signal, and so is available from the start. Then, no theory would have any problem; however, there is no evidence for this assumption.

2. The learner arrives at the representations through some means other than constraint ranking, say by some set of learning principles, P. So, from the initial state, the learner applies P and arrives at the stage which Tesar & Smolensky assume is the input to constraint ranking, call this stage \( S_i \). The questions to ask now are: could the learner have arrived at \( S_i \) without having already ranked the constraints? If no, i.e. if \( S_i \) itself involves constraint ranking, then Tesar & Smolensky’s algorithm is superfluous. If yes, i.e. the learner is at \( S_i \) but has ranked no constraints, then what role do the constraints play? So it seems that the danger is that either the algorithm or the constraints are superfluous. The direction I would pursue is to suppose that \( S_i \) itself involves constraint ranking, i.e. that the establishing of representations and constraint-ranking influence each other, and that both are in motion in the course of acquisition.

5. Conclusion

To conclude, I think that an ordered cue-based learner of the type sketched in (2) is the most promising approach to solving the fundamental problems of grammar acquisition set out in (1). The next step is to attempt to incorporate the results of the work of Fikkert (1994) and others on the actual path of development followed by children. These data show even more forcefully that the target input forms to the learners are moving targets, not given in advance of applying a learning algorithm. Rather, adult representations are mental constructs, themselves the results of the acquisition of grammar.
References