North East Linguistics Society

Volume 19
Issue 1 *NELS 19*

Article 5

1989

On the Relationship between the Input Data and Parameter Setting

Robin Clark
Carnegie Mellon University

On the Relationship between the Input Data and Parameter Setting*

Robin Clark

Department of Philosophy
Carnegie Mellon University

Parameter setting models of language acquisition are an attractive alternative to inductive models. Parameter setting models involve a limited hypothesis space which allows only a finite number of possible target languages. Such a model of Universal Grammar (UG) can contain only a finite number of principles; each principle can involve only a finite number of parameters. Finally, each parameter can range over only a finite number of possible values. Multiplying out the possibilities results in a finite, although potentially large, number of possible target languages which can serve as potential hypotheses for the learner. This number can act as an upper bound on the learning problem. It is difficult to see how models which involve the induction of a set of language particular rules can so easily guarantee such an upper bound on the hypothesis space. Thus, parameter setting models represent an increase in the explanatory power of UG.

North East Linguistics Society, Vol. 19 [1989], Iss. 1, Art. 5

I will argue, here, that a potentially quite difficult problem remains to be accounted for. In particular, given any piece of evidence, how does the learner decide which parameter is the appropriate one to set? I will refer to this problem as the **Selection Problem**. Stated in an alternative way, the selection problem is the problem of discovering the causal relation between some set of input data and setting a parameter.

1. Characteristics of the Learning Problem

I will assume that the learner is presented with examples of simple, grammatical sentences. Each example is presented to the learner who then tries to assign a well-formed linguistic representation to that example, where 'well-formed' is with respect to the learner's current hypothesis grammar. If the current hypothesis grammar can assign a well-formed representation to the example, then the learner simply proceeds to the next example without attempting to reset any parameters. If, on the other hand, the learner cannot assign a representation to the current example, it selects a parameter from the set of available parameters, and sets its choice to a new value. I will assume that the values of each parameter are ordered and that the learner tests new values in sequence.

The basic learning procedure is summarized in (1):

(1) **The Learning Procedure**

(a) Input a string from the example text and parse the string.
(b) If parse succeeds go to (a).
(c) Otherwise, select a parameter and reset it to a new value.
(d) Go to (a).

The above learner has the following properties. First, learning is error-driven. Learning occurs only in the presence of an error; otherwise, the learner's hypothesis remains fixed. Second, the learner has no memory for past examples. When setting a parameter, then, it cannot scan the set of examples it has already seen to test if its new hypothesis will be consistent with these examples. Third, the learner can only reset one parameter at a time. As a result, it can revise its hypothesis only one way per error.[1]

Following much recent work (Berwick, 1985; Wexler & Manzini, 1987; among many others), I will assume that the languages generated by various parameter settings can array themselves in various set-theoretic ways. One possibility is that the languages generated by two distinct settings of a parameter fall into intersecting or completely disjoint sets. I will refer to the former as **intersecting languages** and the latter as **disjoint languages**. Crucially, I will further assume that the languages generated by distinct values of a single parameter can fall into the subset relation. I will refer to this case as **subset** or **superset languages**.

In the case of subset languages, the learner must hypothesize the value that generates the smallest language first and then proceed in increasing orders of magnitude (Berwick, 1985; Wexler & Manzini, 1987; among others). For concreteness, I will assume that the set-theoretic relations that parameter values generate are invariant with respect to the settings of other parameters.[2] In the case of intersecting and disjoint languages, the learner will always have at least one grammatical sentence to which it cannot assign a well-formed representation. Thus, there is always a chance that the learner will encounter the relevant type of example that will force it to revise its current hypothesis.

Notice that the above learning procedure does not specify a method for how the learner goes about selecting a parameter to reset. The problem of specifying this method for selecting a parameter will be addressed in the following sections.

2. A Stochastic Learner

The simplest method for selecting a parameter would be to randomly select one from the list of available parameters and set it to its next value. I will refer to this type of learner as a **stochastic learner**. When the learner fails to assign a well-formed representation to some input, it initiates a learning sequence which selects a parameter on the basis of some random process. For example, the learner could generate a random number and use that number to make a selection from the set of parameters.

A stochastic learner will succeed in learning the target grammar where all parameters generate either intersecting or disjoint languages since in both of these cases, if the learner has made a mistake, there is at least one sentence in the target language that is not contained in the language generated by the hypothesis grammar. If, however, the set of parameters contains at least one whose values generate subset languages, then a stochastic learner cannot be guaranteed to converge on the correct target.

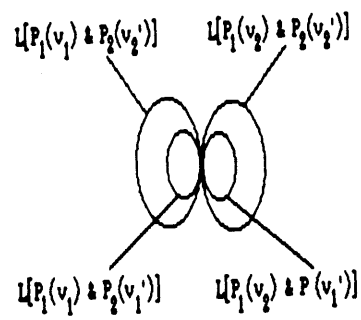
To illustrate this problem, consider the following "toy" model of UG. Let us suppose that UG contains only two parameters P_1 and P_2. Suppose that P_1 has as its possible values v_1 and v_2 while P_2 has as its possible values v_1' and v_2'. Suppose further that the language generated by setting P_1 to v_1 (I will note this as "$L[P_1(v_1)]$) is a not subset of a the language generated by setting P_1 to v_2. That is, the languages generated by setting the values of P_1 are either intersecting or disjoint. The values for P_2, on the other hand, do stand in a subset relation. The relevant subset relations are shown in (2):

(2) a. $L[P_1(v_1)$ & $P_2(v_1')]$ is a subset of $L[P_1(v_1)$ & $P_2(v_2')]$

b. $L[P_1(v_2)$ & $P_2(v_1')]$ is a subset of $L[P_1(v_2)$ & $P_2(v_2')]$

Graphically, the set of languages are shown below:

(3)



Suppose that the learner's initial state has P_1 set at v_1 and P_2 set at v_1' and that the target grammar has P_1 set at v_2 while P_2 remains set at v_1':

(4) a. Initial State: $[P_1(v_1)$ & $P_2(v_1')]$

b. Target State: $[P_1(v_2)$ & $P_2(v_1')]$

In order to successfully discover the target language, the learner need only move to the disjoint (or intersecting) language generated by P_1.

Now consider the following possible learning sequence. The learner is presented with an initial datum, d_1, which it then attempts to parse. Suppose that the learner fails to assign a representation to d_1 so that the learning sequence begins. The random procedure selects a parameter to set, in this case the learner selects P_2 which is set to its next value, v_2'.

Notice that the learner has jumped to a superset language; $L[P_1(v_1)$ & $P_2(v_2')]$ is a superset of $L[P_1(v_1)$ & $P_2(v_1')]$. Since the target language is generated by P_1, it is still disjoint (or intersecting) with the current hypothesis language. As a result, there are still strings in the target language which the learner cannot process. Suppose that the next datum, d_2, is one such example. The learner receives d_2 and fails to assign it a representation and the learning sequence is again initiated. This time the random procedure selects P_1 which is set to its next value, v_2.

At this point, the learner has entered a superset language to the target language. Its current hypothesis language, $L[P_1(v_2)$ & $P_2(v_2')]$, properly contains the target language, $L[P_1(v_2)$ & $P_2(v_1')]$. Now, any further data the learner encounters will be consistent with its hypothesis, since the target language is properly contained in the hypothesis language. Since learning is error-driven, the learner will never be forced to revise its hypothesis and will never discover the correct target grammar.

The intuitive reason that the stochastic learner can fail, even in so simple a model, is that it cannot relate properties of the input string to the set of available parameters. That is, making a certain kind of error should cause the learner to select a parameter that can help prevent that type of error from reoccurring.

Suppose that the learner misanalyzes an Exceptional Case Marking (ECM) structure (Chomsky, 1981) as (5a) rather than (5b):

(5) a. $[_{IP}$ John $[_{VP}$ believes $[_{CP}$ $[_{IP}$ Bill to be late]]]]

b. $[_{IP}$ John $[_{VP}$ believes $[_{IP}$ Bill to be late]]]

In structure (5a), the subject of the embedded clause, "Bill," cannot receive Case since the embedded clause is untensed and the superordinate verb, "believe," does not govern it. Ideally, the learner should recognize the source of the problem and make some appropriate change to its current hypothesis. For example, the learner could localize the problem to the theory of government and adopt an [+ECM] hypothesis.

A stochastic learner cannot make the appropriate relations between the input data and the set of parameters. It is clearly useless for a learner to manipulate a parameter in the binding theory when confronted with the structure in (5a). Such a manipulation would do no good and could, potentially, do great harm. Nevertheless, nothing prevents a stochastic learner from making such bad choices.

3. A Counterfactual Learner

The previous section provided a short demonstration of the necessity of specifying the causal relation that must exist between the input data and the process of selecting a parameter to reset. If we cannot account for this relationship in our theory of language acquisition then the theory is suspect.

An obvious method for repairing the learner is to allow it to search the set of parameters until it finds one that will allow it to assign a representation to the input datum.[3] That is, the learner could attempt to support the following counterfactual statement:

(6) If parameter P_i is set to value v_j, then input datum d_k receives a well-formed representation.

In other words, the learner searches around the parameter space until it finds a setting that will help solve its current problem. It will only reset those parameters that actually help it circumvent error. The learner's behavior should appear much more purposive, under this model.

The counterfactual model, however, is subject to certain complications. Consider the contrast in (7) and (8) (the examples in (8) are from Chung & McCloskey, 1987):

(7) a. John believes [$_{IP}$ Bill to be ill]

b. They$_i$ believe [$_{IP}$ each other$_i$ to be ill]

(8) a. Is cuimhneach leo [iad a bheith
Cop mindful with-them them be([-Fin])

ar seachran]
lost
"They remember being lost."

b. *Shil siad [a cheile a bheith
think(Past) they each-other be([-Fin])

breoite]
ill
"They thought that each other was ill.

The examples in (7) are standard English ECM structures. Notice that an anaphor in the subject position of the embedded clause is licensed in English (example (7b)). Since the superordinate verb governs the embedded subject position, the governing category of the subject shifts to the superordinate clause. Thus, ECM interacts with the binding theory in non-trivial ways (Chomsky, 1981).

Chung & McCloskey (1987) argue that Modern Irish allows for the Case-marking of structural subjects in the absence of a governor. Thus, although Modern Irish lacks true ECM structures, a phonologically overt subject can be licensed quite generally by means of structural Case-marking. Notice that, since Irish lacks ECM, an overt anaphor is not licensed in the subject position of an embedded non-finite clause, as shown in (8b). This follows since the anaphor will not be governed by the superordinate verb and, hence, will not have the superordinate clause as its governing category.

When presented with an example like (7a), how can the learner distinguish between ECM-type languages (English) and structural-Case-Marking-type languages (Modern Irish)? The parameters involved are quite distinct: The ECM parameter may be in the government component of the grammar while the structural Case-marking parameter is in the Case theoretic component of the grammar. The learner must be sensitive to interactions between several distinct parameters.

To control for this type of interaction, the learner would have to consider predictions made by possible parameter settings. To see how complex this might be, consider the logical possibility that the target grammar allows for structural Case-marking (like Irish) and long-distance anaphora (like Icelandic). Such a language would be a superset of an ECM language like English in that it would allow examples like (7b) (using a long-distance anaphor) as well as having a free distribution of phonologically overt subjects in untensed clauses.

Thus, it is possible that superset languages can be generated not only with respect to the values of a single parameter but also with respect to combinations of parameters. The learner must be sensitive to the possible interactions between parameters in order to avoid accidentally entering a superset language.

The complexity that the learner faces is inevitable in a system where the parameters record more than mere taxonomic properties of natural language. Instead, the parameters interact to generate complex linguistic patterns. The learner faces the problem of unpacking these complex interactions in order to discover the correct parameter settings for the target grammar. The problem here is that of discovering how a learner can disentangle these complexities automatically in a relatively small amount of time. The difficulties here are considerable and resemble the **Frame Problem** found in Artificial Intelligence (see McCarthy & Hayes, 1969; Fodor, 1987 provides an accessible discussion).

For present purposes, the problem is that the learner must test for interactions between sets of parameters. At some point, it must have an effective procedure for telling it that it can stop testing for adverse consequences and can set a parameter. Even if such a procedure could be developed, it is unlikely that a counterfactual learner would be at all efficient at its task due to the sheer enormity of the search space now involved. Recall that the learner must consider the interaction of sets of parameters; this could lead to a combinatorial explosion of possible counterfactual statements that must be considered even if the number of parameters is relatively modest.[4]

I will therefore put aside counterfactual learners to pursue more promising approaches to the structure of parameter setting.

4. Minimization

A minimizing learner would, at every step of the learning procedure, hypothesize the smallest language compatible with the input data. Given a new piece of input data, the learner would scan the set languages that differ from the current language by one parameter value.[5] If a set of languages is compatible with the current input, the learner would select the least language of that set (the most restrictive parameter setting possible) and reset the relevant parameter to generate that language.

The minimizing learner, then, searches for parameter settings that will allow it to eliminate the current error (like the counterfactual learner), but its search is more constrained. It makes the most restrictive hypothesis it possibly can, given the input. In this sense, it seems to encode the Subset Principle (Berwick, 1985; Wexler & Manzini, 1987).

A detailed critique of the minimizing learner is beyond the scope of the present paper (see Clark, in prep, for a detailed criticism of this model). For current purposes it is sufficient to note the following two comments. The first point is that, although the search space that the minimizing learner must consider is constrained, it could still be quite large. Every language generated by a parameter setting which differs from the current setting by one value must be considered. The learner must then compute the relative inclusion relations of the languages. The computational demands on the learner are far from trivial. Secondly, this type of learner makes few concrete predictions about the actual time course of learning. To the degree that the computational theory of learning should illuminate developmental data, this is a serious weakness in the model.

5. A Sequential Learner

Both the counterfactual learner and the minimizing learner have access to all parameters at any given time. Both learners attempt to model the causal relation between input data and parameter selection by considering the semantic properties of the parameters. An alternative route would be to model parameter setting by depriving the learner of access to certain parameters. In essence, parameter selection is predetermined by a fixed schedule so that the learner is never forced to select between rival hypotheses. I will call this type of learner a **sequential learner.**

The parameters in the sequential model are strictly ordered so that P_1 precedes P_2 and so on. The first parameter, P_1, is made available to the learner and, after some amount of time, becomes unavailable. If the learner makes an error on some datum during this period, it will select a new value for P_1. This process continues throughout the ordering until the final parameter, P_n, is set. At no point is the learner free to select between two different parameters. Thus, this model mimics causal relations between the input data and parameter setting by means of the syntactic device of linear ordering.

As it stands, however, this model will allow irrelevant mistakes to trigger parameter setting. It is possible, for instance, for an error due to the learner's current hypothesis about bounding theory to trigger a change in the Case component of the grammar. Due to the syntactic nature of the learning procedure, the learner simply changes the setting of the parameter currently available to it when it notes that mistakes were made.

In order to circumvent this problem, we can propose that the learning procedure has a special component which filters out certain types of errors and allows other errors to pass unnoticed. As an example of a filter, at some stages the learner may not attend to words to which it cannot assign a regular semantic function. Thus, the learner will fail to attend to pleonastic elements since these lack referential content. As a result, sentences containing pleonastic elements will not generate errors at these stages. If, following Hyams (1986) pleonastic elements are part of the triggering data for resetting the Null Subject Parameter (NSP), then there should be a significant period where the learner fails to reset the NSP despite the available counterevidence. Since the pleonastic elements have been filtered out of the input, examples like (9) will simply not generate an error for a learner at this stage:

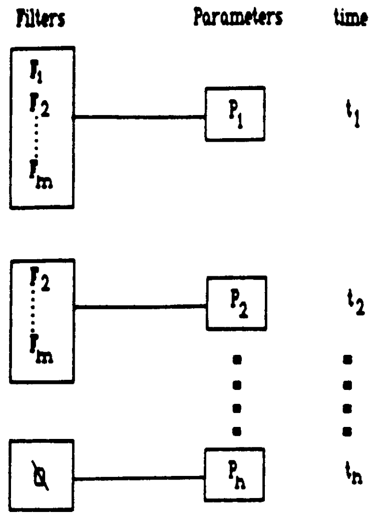
(9) There is a man in the park.

At a later point, the pleonastic filter will be discarded and examples like (9) will generate errors and trigger parameter setting.

Equally, certain constraints on representations may be inactive at some stages; any representation which violated such a constraint would not generate an error for the learner. If, for example, the Case component were inactive at early stages of development, then the Case Filter would not be violated in structures where no abstract Case is assigned. One would expect the learner to accept simple subject-predicate structures which syntactically resemble small clauses and in which the subject has not been assigned Case (cf, Kazman, 1988). Equally, the learner may be unable to represent certain relations (e.g., A-chains) as Borer & Wexler (1987) have argued. The filters, then, both direct the attention of the learner to certain types of examples and govern the representational capacities of the learner.

The filter component, then, acts to direct the learner toward certain types of errors, namely those that are relevant to the parameter which the learner must currently set. The basic form of the sequential learner is shown in (10):

(10)



At each point of time, a single parameter is available. The filter component adjusts the data available to the learner so that relevant examples generate errors while irrelevant data is ignored. In addition, the filter component governs the representational capacities of the learner. Each parameter is associated with a packet of filters; when the parameter is no longer available, the packet associated with it is deleted so that new types of examples will generate errors.

The sequential learner is explicitly maturational in the sense of Borer & Wexler (1987); I would argue, in fact, that there are sound computational reasons for maturation. A maturational theory prevents the learner from becoming incapacitated by the sheer enormity of the possible hypotheses that could account for the input data. In addition, such a learner provides a rationale for the often noted stages of acquisition (Brown, 1973 being a classic discussion of developmental regularities). Stages should correspond to the availability of parameters. Thus, it should come as no surprise that children pass through regular stages during the acquisition of their mother tongue.

Endnotes

* Portions of this paper are to appear in Behavioral and Brain Sciences. I have benefitted from helpful discussions with Clark Glymour, Tom Roeper and Ken Wexler.

[1] See Wexler & Culicover (1980) for a discussion of error-driven learning and for a discussion of intensional vs. extensional learners. This latter distinction deals with the encoding of past examples either via direct memory (extensional learners) or in the form of a grammar (intensional learners). Finally, see Berwick (1985) for a discussion of recursive calls to the learning routine.

[2] This is the Independence Principle of Wexler & Manzini (1987). It prevents a subset language of one parameter from becoming a superset language based on the setting of some other parameter. If this situation ever arose, the learner could inadvertently enter a superset language to the target language. Since the data is assumed to be positive exemplars from the target language, all further examples the learner sees would be consistent with its hypothesis. But the learner fails to converge to the correct grammar. See the discussion in Berwick (1985).

[3] It is possible that the current input is ill-formed in several ways. One might imagine, then, that no one resetting of a parameter would "fix" the problem. To remedy this, one might allow the learner to search the parameter space until it finds a setting that minimizes the deviance of the current input. Defining such a procedure is far beyond the scope of the present paper. Alternatively, the learner might abandon hope until it encounters a simpler triggering datum.

[4] To get an idea how large, consider the number of chess games. Although the number of ways a chess piece can move at any one time is quite restricted, the number of possible games is estimated (in Simon, 1982) to be on the order of 10^{120}. One might expect the number of possible interactions between parameters to be enormous.

[5] This learning procedure was suggested to me by Clark Glymour. Recall that the learner has no memory of past examples; the language seen to date is indirectly encoded by means of the grammar hypothesized by the learner (see endnote 1). Furthermore, by hypothesis above, the learner can only set a single parameter at a step; as a result, the set of languages that the learner

scans in selecting a new hypothesis grammar need only be those generated by parameter settings that deviate from the current state by one value.

References

Berwick, R. (1985). The Acquisition of Syntactic Knowledge. The MIT Press, Cambridge, MA.

Borer, H. & K. Wexler (1987). "The Maturation of Syntax" in T. Roeper & E. Williams (eds) Parameter Setting. D. Reidel Publishing Co., Dordrecht, the Netherlands.

Brown, R. (1973). A First Language: The Early Stages. Harvard University Press, Cambridge, MA.

Chomsky, N. (1981). Lectures on Government and Binding. Foris Publications, Dordrecht, the Netherlands.

Chung, S. & J. McCloskey (1987). "Government, Barriers and Small Clauses in Modern Irish". Linguistic Inquiry, 18.2, pp. 173-237.

Clark, R. (1989). "Causality and Parameter Setting". Behavioral and Brain Sciences, 12.2.

Clark, R. (in prep). "Four Models of Parameter Setting." CMU & UCLA.

Fodor, J. A. (1987). "Modules, Frames, Fridgeons, Sleeping Dogs, and the Music of the Spheres" in J. Garfield (ed.) Modularity in Knowledge Representation and Natural-Language Understanding. The MIT Press, Cambridge, MA.

Hyams, N. (1986). Language Acquisition and the Theory of Parameters. D. Reidel Publishing Co., Dordrecht, the Netherlands.

Kazman, R. (1988). "Null Arguments and the Acquisition of Case and Infl". paper presented at the Thirteenth Annual Boston University Conference on Language Development.

McCarthy, J. & P. Hayes (1969). "Some Philosophical Problems from the Standpoint of Artificial Intelligence" in D. Michie (ed.) Machine Learning 4, Edinburgh University Press.

Simon, H. (1982). Models of Bounded Rationality: Behavioral Economics and Business Organization; Volume 2. The MIT Press, Cambridge, MA.

Wexler, K. & P. Culicover (1980). Formal Principles of Language Acquisition. The MIT Press, Cambridge, MA.

Wexler, K. & M.-R. Manzini (1987). "Parameters and Learnability in Binding Theory" in T. Roeper & E. Williams (eds) Parameter Setting. D. Reidel Publishing Co., Dordrecht, the Netherlands.