To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

Computational Approaches to Parameter Setting in Generative Syntax

William Gregory Sakas

Ph.D. Programs in Linguistics and Computer Science, The Graduate Center
Department of Computer Science, Hunter College
CUNY – The City University of New York

1 Introduction

1.1 Principles and Parameters

Since the very beginning of modern generative linguistics, a central tenet of the field has been that any theory of human language grammar must provide a viable account of how a child[1] language learner acquires the grammatical knowledge of an adult. In *Aspects of the Theory of Syntax*, Chomsky (1965) provided one of the earliest generative accounts of the process. His acquisition model consisted of two components: a rule writing system and a formal evaluation metric. The process of acquisition was envisioned as the child's construction of a grammar by creating rules that were compatible with the linguistic environment they were exposed to, and given a choice of grammars, selecting the grammar most highly valued under the evaluation metric.

At the time, most all generative theories of syntax were highly transformational. Although both simplicity and economy were proposed as the proper evaluation metrics, transformational rules were often highly complex and unconstrained making it difficult to form a working model of how a simplicity or economy evaluation metric could be applied.[2] The *principles and parameters* framework (Chomsky, 1981, 1986) grew out of the inability of linguists to bring to fruition this early vision of language learning.

The principles and parameters framework does away with both the rule writing system and the evaluation metric. Instead the *principles* are the grammar constraints and operations that govern *all* possible human languages, and *parameters* are the points of variation *between* languages. Learning is the process of setting parameters to exactly the parameter values that generate the child's native

[1] Throughout I mean 'child' and 'children' to be inclusive of infant- and toddler-aged young humans.
[2] There were parallel concerns related to how features and markedness in phonology could be reconciled with an evaluation metric, see discussion in *The Logic of Markedness* (Battistella, 1996).

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

language.[3] Once the parameters are set to their correct values, the child has achieved adult syntactic competence. Both the principles and parameters are prescribed innately as part of Universal Grammar (UG). As Chomsky puts it:

> We can think of the initial state of the faculty of language as a fixed network connected to a switch box; the network is constituted of the principles of language, while the switches are options to be determined by experience. When switches are set one way, we have Swahili; when they are set another way, we have Japanese. Each possible human language is identified as a particular setting of the switches—a setting of parameters, in technical terminology. If the research program succeeds, we should be able literally to deduce Swahili from one choice of settings, Japanese from another, and so on through the languages that humans acquire. The empirical conditions of language acquisition require that the switches can be set on the basis of the very limited properties of information that is available to the child. (Chomsky, 2000, p. 8)

A grammar, in current theories of generative syntax, is no longer construed as a set of rules and transformation operations, but rather as a vector of labeled *parameters* together with *parameter values* that generate a particular human language. As an example: all sentential clauses in all languages must have a subject by the Extended Projection Principle (EPP). Whether or not the subject is obligatorily explicit is determined by a parameter, the so-called 'Null Subject' parameter, together with its parameter value. English has the Null Subject parameter set to the value *–Null Subject*, and Spanish has it set to the value *+Null Subject*. The number of parameters that define the domain of human grammars (the number of switches in Chomsky's metaphorical description above) is posited as finite and the parameters are standardly considered to be binary, i.e., there exist two (and only two) values for each parameter. Although this description of the Null Subject Parameter lives in Government and Binding Theory, grammars in the Minimalist program are also parameterized; parameters are feature-based and housed in the lexicon.[4]

At first blush it would seem that the principles and parameters framework (hereafter P&P) offers a much more manageable account of acquisition than the model Chomsky first proposed in *Aspects*. Instead of generating a rule from a virtually infinite (i.e., unconstrained) space of rules and applying an evaluation metric to an infinite set of grammars, all a child needs to do is to observe the relevant linguistic characteristic of a single utterance (e.g., a declarative sentence without a subject), and some

[3] Throughout I mean 'native language' to be inclusive of native languages in the case that a child is brought up in a multi-lingual environment, and likewise for 'target grammar'.

[4] For reasons of narrative and space, discussion here is largely restricted to computational acquisition models of binary syntactic parameters. It's important to note that there are important computational results stemming from research in P&P acquisition that use multi-valued (e.g., Manzini & Wexler, 1987; Wexler & Manzini, 1987; Briscoe, 2000; Villavicencio, 2001), and models that address P&P acquisition of phonology rather than of syntax (Dresher & Kaye, 1990; Dresher, 1999; Pearl, 2007).

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

parameter can be set appropriately (e.g., to +Null Subject). Indeed, though only lightly sketched out when the P&P framework was conceived, it was widely accepted that this 'switch-flipping' concept of parameter setting was how language acquisition proceeded. Parameter setting was more or less instantaneous, 'automatic', accurate and deterministic. There appeared to be no need to search a space of grammars, and no need to revise a parameter value once a choice was made. This view has often been referred to as *triggering theory*, *triggering* or the *trigger model* of acquisition.

Most (though not all) computational models of parameter setting have taken another approach. The acquisition process is implemented as a *nondeterministic* (trial-and-error) search of the *grammar* or *parameter space* (all possible combinations of parameter values, i.e, all possible grammars). These approaches are the antithesis of the triggering approach which is about the *deterministic* (non-trial-and-error) process of setting parameters based on reliable evidence; triggering has no need to conduct a search of the grammar space – given proper evidence to trigger one parameter value over another, set the parameter to that value once and for all.

The computational psycholinguistics community has, by and large, abandoned triggering theory. This has come about not for lack of interest in triggering, but rather because of the realization that the domain of human languages is vexed with a considerable amount of *parametric ambiguity*. There are many examples of syntactic phenomena which can be licensed by multiple combinations of parameter values necessitating that the learner choose from amongst alternatives (Clark, 1989, 1992)[5]. Though the correct parameter values are necessarily among the alternatives, there would presumably be many incorrect alternatives. If true (at least as far as the argument goes) a triggering model in the best case would be stymied into inaction due to ambiguity, in the worst, it would make unrecoverable errors. In short, parametric ambiguity is the driving force behind the genesis of computational approaches to parameter setting and will serve well as a central theme throughout this chapter. In what follows I present a brief overview of three core concepts relevant to the computational modeling of language acquisition, then a non-exhaustive history of research specifically on computational models of P&P acquisition, and finally draw together some points on whether or not the deterministic triggering theory is viable after all.

1.2 No Negative Evidence, Learnability vs. Feasibility, and the Subset Principle

The term negative evidence in the study of language acquisition refers to evidence about what is not grammatical in the *target language* (the language learner is being exposed to). The evidence can take many forms: overt statements from caregivers about what isn't grammatical, repetitions of a child's ungrammatical utterance but with a minor grammatical change, urging by caregivers to get a child to repeat ungrammatical utterances but grammatically, etc. [6] Most accounts of acquisition in a generative

[5] A concrete example is given in the section *Clark and Parametric Ambiguity*.

[6] Another type of negative evidence is statistical in nature. If a linguistic construction never occurs in the input, at some point, it could be argued, that a construction in ungrammatical. This type of negative evidence is referred to

framework accept results from psycholinguistic research (e.g, Brown & Hanlon, 1970; Marcus, 1993) that children do not receive negative evidence. All computational modeling endeavors in P&P acquisition make this assumption as well – only sentences from the target language are encountered by the learner, and there is no 'extraneous' information about ungrammaticality.

Learnability, often referred to as the "logical study of language acquisition", attempts to answer the question: Under what conditions, in principle, is acquisition possible? There is an important distinction to be made between learnability and *feasibility* (Kapur, 1994; Sakas, 2000). Feasibility attempts to answer a different question: Is acquisition possible within a reasonable amount of time and/or with a reasonable amount of computation? Though this second question has been raised even before P&P (e.g., Pinker, 1979), its importance has increased post-P&P since *any* P&P domain consisting of a finite number of parameters and values is formally learnable (Osherson, Stob, & Weinstein, 1986; Bertolo, 2001). By "formally learnable", I mean that there exist mathematical proofs establishing that there is *some* learner that will acquire every language in any and all P&P domains. The proofs demonstrate (given some assumptions[7]) that an exhaustive search of any finite space of grammars (P&P or not) will eventually discover the *target grammar* - the grammar generating the sentences of the language the learner is being exposed to. Since a grammar space delineated by a P&P framework is by definition finite, any P&P domain is learnable.

However in the case of a P&P domain, the domain quickly becomes quite large; a mere 30 parameters yields a space of 2^{30} grammars – over a billion (actually exactly a gigabyte of grammars – well over the 3,819,816 possible combinations of 5 numbers on the Mega Millions Lottery), and grows exponentially in the number of parameters. 40 parameters would delineate a space of 2^{40} = over a trillion grammars (more than the number of neurons in the human brain), and 50 parameters would manifest a grammar space equal to the number of stars in five thousand Milky Way galaxies (each galaxy containing approximately two hundred billion stars). Since children presumably don't embark on exhaustive search over multiple grammars on every input sentence, learnability results concerning (finite) P&P domains are unattractively weak from the point of view of computational psycholinguistics. Still, both questions are relevant. There are psychologically attractive computational models that are not able to acquire all the target languages in a domain (i.e., which don't meet learnability criteria). And models which do acquire all the target languages[8] but require an unwieldy number of input sentences (i.e., which don't meet feasibility criteria).

---

as *Indirect negative evidence* (Chomsky, 1981). Indirect negative evidence plays a particularly important in Bayesian statistical models of acquisition, but doesn't play a role in the P&P models discussed in this chapter.

[7] Most notably the consideration by the learner of grammars of that generate subset languages *before* grammars that generate their superset languages. See discussion immediately below.

[8] Although learners in all generative theories of syntax *acquire* a *target grammar*, the exposition is sometimes clearer when the phrase *acquire a target language* is used. The reader should take *acquire a language* to mean *acquire a grammar*.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

The *Subset Principle*, first proposed by Gold (1967) (though not by that name), and later revisited in depth from more linguistic viewpoints (Berwick, 1985; Manzini & Wexler, 1987; Wexler & Manzini, 1987) is a product of formal learnability theory which has significant implications for theories of human language learning. An informal definition of the Subset Principle will suffice here: *the learner must never hypothesize a language which is a proper superset of another language that is equally compatible with the available data*. The basic idea is that if a language learner were to hypothesize a superset of the target language, he or she would presumably be forever caught in the superset language unable to retreat to the target language. This is because the only evidence available to the learner are sentences of the (subset) target language (given the assumption that there is no negative evidence), all of which are compatible with the incorrectly hypothesized (superset) language. In theory, a child who is unfortunate enough to hypothesize a superset of his or her target language would become an adult able to comprehend all utterances of their fellow native speakers, but since their internal grammar licenses the superset language they would presumably be uttering ungrammatical sentences that are not licensed by the subset grammar. Since human learners do not exhibit this behavior [9], it has generally been accepted that some form of the Subset Principle must be followed by the child language learner.

However, many issues concerning the Subset Principle are currently unresolved: Do children actually apply the Subset Principle? Perhaps children violate the Subset Principle and are yet able to retreat to the subset target language? Do subset-superset relationships actually exist in the domain of human languages? How could children apply the Subset Principle incrementally (in real-time) without global knowledge of what sentences are licensed by each and every grammar in the human language domain? Extensive discussion of these points can be found in Fodor and Sakas (2005), but for purposes here, we only need to keep an eye on how each of the computational models that are reviewed deals with, ignores or sets aside issues related to subset-superset relationships between languages.

2 The Models

What follows is a review of computational models of parameter setting post Chomsky's original conception of the process. The reader wishing to skip to one particular model is advised to first read the following longish section *Clark and Parametric Ambiguity* as it introduces a number of concepts that have prevailed in the field for over two decades. Discussions following the section on Clark will make reference to points made there. This review is certainly not comprehensive, but at least in my opinion the models mark important stages in computational approaches to parameter setting, and taken together tell a comprehensible story.

[9] Though there is the occasional report in the literature of overgeneration and subsequent retreat by children (Deprez & Pierce, 1993).

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

2.1 Clark: Parametric Ambiguity

One of the earliest computational models of parameter setting was presented by Robin Clark (Clark, 1989, 1992). Importantly, Clark was the first to point out that any parameter setting model of language acquisition would need to address *parametric ambiguity*: when syntactic phenomena can be licensed by multiple combinations of parameter values which would necessitate the learner choosing from amongst alternatives. Clark presents extended discussion of ambiguity involving how case theory, θ-theory, government theory and lexical properties interact to create a variety of parameter settings – meaning different grammars – that would license a subset of English sentences. But a simple example will suffice here.

A surface sequence of *Subject Verb Object* may either be generated as a base structure with no movement (as in English) or be derived from a base structure such as *Subject Object Verb* by movement of the verb to the second position (as occurs in German). One syntactic description of this *V2* phenomenon involves three proposed parameters: Obligatory Topic (ObT), V-to-I movement, and I-to-C movement. For each parameter a + parameter value indicates 'obligatory'. +ObT dictates some topicalizable lexical element (or constituent) be moved into Spec,CP, +V-to-I dictates movement of a tensed verb (or aux) under I from its base-generated position dominated by V, and +ItoC dictates movement of a tensed verb (or aux) from I to C. Under this description, German is +ObT, +V-to-I and +I-to-C. The surface effect is that the topic (in Spec,CP) is the first realized item in the surface string, and the verb, moved from V to I then from I to C, is necessarily the second item in the string. The result is that the same string can be assigned at least two structures licensed by UG.[10,11]

What does this mean to the child language learner? Consider a child encountering the sentence *Mary sees the dragon*. If the child were to (mistakenly) assign the German parameter values she could parse the sentence and interpret it, but would presumably at some point utter a sentence such as *John will the ball hit* which is licensed by the German, but not by (contemporary) English. Since there are very few errors of comission made by children (Snyder, 2007) it would seem that the parameter setting mechanism must either be:

i) recognizing the ambiguity and choosing not to adopt either the German or English settings,[12] or,
ii) ignoring the ambiguity, guessing a grammar hypothesis – recall, a vector of parameter values (perhaps monitoring how well each value performs on sentences encountered in the future). At some point during the course of acquisition, the learner stops guessing and acquires the correct parameter values and uses them for production.

[10] Note that it could be licensed by other combinations of parameter values as well, for example by optional topicalization of S (-ObT), and, V in root position (-VtoI) or V in I position (+VtoI) without movement into C (-ItoC) .
[11] This is also an example of what Clark calls *parametric interaction*: when parameter values conspire to generate surface phenomena that do not reveal their parametric signature. I.e., a learner given a surface string can't tell which parameters (or parameter values) were employed in generated the string.
[12] Though perhaps temporarily adopting one or the other for purpose of understanding the sentence at hand.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

Clark also defines a related difficulty parametric learners are faced with, that of *parametric expression*. As an example consider another proposed parameter: *the pied piping parameter*. A + value of this parameter dictates that the complement of a prepositional phrase is moved with its head. A – value allows the complement to move and leave the head behind. French is +pied piping, Colloquial English is –pied piping. (The syntactic details of the parameter are not important for present purposes.) Now, *Mary sees the dragon* doesn't contain a prepositional phrase – neither value of the pied-piping parameter is expressed. As a result the sentence can be parsed with either value. This is different from parametric ambiguity since the setting of the pied piping parameter is irrelevant to the structure of the sentence.

In the case of pied piping, it might seem fairly easy to recognize which parameters a sentence expresses, but that's not true of all parameters. Take the ObT parameter, if the subject (*Mary*) has moved into topic position in *Mary sees the dragon* then in order to determine if the +ObT value is expressed, the learner would need to know if the movement was due to a principle of core grammar (all grammars allow topicalization) or due to a +ObT setting. If *Mary* remains in the base-generated subject position (Spec,IP) then the –ObT parameter value is clearly expressed[13] since *Mary* is at the head of the sentence nothing can have been moved into topic position (higher than *Mary*). To determine if the ObT parameter is expressed in the utterance, the learner would need access to the grammar that was used to produce the utterance by the speaker which she clearly does not possess. This is also true for the verb movement parameters discussed above. Not only is parametric expression related to ambiguity, it intertwined with it in non-trivial ways.

Finally, a point unrelated to either parametric expression or ambiguity. Clark notes that learning should be constrained so that the acquisition process proceeds gradually. I.e., that the learner doesn't hypothesize wildly different sets of parameter values after receiving each input sentence. To encode this notion, Clark defines the *Single Value Constraint*. "… language learning is gradual by specifying that each successive hypothesis made by the learner differs from the previous hypothesis by the value of at most one parameter." (Clark, 1992, p. 90). Although the computational approach employed by Clark doesn't require the Single Value Constraint, it deserves mention as it has been adopted by Gibson & Wexler's TLA model described below. (Gradualism is discussed at length by Yang (2002), see Section 2.4 below).

Given the difficulty of identifying and resolving the potentially complicated entanglements associated with parametric ambiguity, Clark was the first computational linguist to relinquish triggering and to employ a standard machine learning approach from artificial intelligence in his model of parameter setting, a *genetic algorithm*. A genetic algorithm (GA) is a search strategy designed to mirror evolution. In its most generic form the algorithm receives an input, then the GA considers a pool of potential hypotheses (a population) by applying a *fitness metric* to all the hypotheses in the pool. After the fitness

[13] Note that this is not so clear if the grammar allows null topics. Then there may have been a topicalizable element (e.g., an adverb) moved into spec,CP and subsequently deleted. In this case it's *un*clear if the –ObT parameter is expressed. See discussion of these sorts of parameter interactions in Sakas and Fodor (Sakas & Fodor, Submitted).

metric is applied, the least fit hypotheses are removed from the pool ("die off"), the attributes (genes) of the most fit are combined (mating), and some attributes are changed at random (mutation). The process repeats until some termination criterion (or criteria) is met; only the most fit hypotheses survive. [14]

Genetic algorithms implement a form of *heuristic search* – a strategy that is expected to identify the optimal hypotheses in the search space (e.g., the space of all possible parametric grammars), but is not logically *guaranteed* to converge on an optimal solution. Heuristic search is most often employed when the search space is too large and unruly for efficient implementation of an algorithm that would provably converge, e.g., exhaustive search. However in the case of modeling human learning, psychological plausibility is also a consideration. Clark speculates (p. 143) that his GA model could reasonably be embodied in a human language learner which makes it an attractive alternative to exhaustive search (see discussion above in the Section 1.2) though there has been some critique (see below). Indeed all the search strategies outlined in this chapter that have been suggested for models of parameter setting are heuristic.

In Clark's parameter setting model the pool of hypotheses are parameterized grammars; each hypothesis is a vector of parameter values (a string of +'s and –'s), each value is 'gene'. When an input sentence is encountered the algorithm parses the sentence with all the grammars in the pool. Then a particular fitness metric is applied to all the grammar hypotheses. In its calculations, the fitness metric prefers:

1) Grammars that return fewer parsing violations
2) Grammars that generate subset languages
3) Grammars that require fewer parameter values to successfully parse the input

After a fitness ranking is calculated for all the grammar hypotheses in the population, Clark's GA proceeds as described above.

Given:

- a set hypotheses, *Pool* = (h_1, h_2, …, h_n), i.e., a set of grammars,
  where each h_i = (p_1, p_2, …,p_m), each p_i is a either a + value or a – value
- a set of fitness values, *Fit* = (f_1, f_2, …, f_n) where each f_i is the fitness of hypothesis h_i based on 1), 2) and 3) above
- a function *MAX* which randomly selects two hypotheses probabilistically favoring the most fit hypotheses over the less fit

[14] The terminology may become a little confusing due to the use of both standard GA jargon and linguistically appropriate terminology based on context. A mini thesaurus might help:
population = pool = collection of hypotheses
member of the population = hypothesis = grammar = vector of parameter values = string of +'s and –'s
gene = attribute = a single parameter value of a grammar.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

a sketch of the algorithm follows.[15]

Clark's Genetic Algorithm

- Set *Pool* to *n* randomly chosen hypotheses
- For each input sentence *s*:
    - For each hypothesis h_i in *Pool* :
        - Parse *s* with h_i
        - Calculate each hypothesis' fitness, f_i
    - Remove the least fit hypotheses from *Pool*
    - If *Pool* contains only one hypothesis: STOP
    - Apply *MAX* to identify h_{max1} and h_{max2}
    - Mate h_{max1} and h_{max2} to generate two new hypotheses, h_{child1} and h_{child2}
    - Add h_{child1} and h_{child2} to *Pool*
    - Randomly pick a single hypothesis, perform mutation.

During the course of learning, the fitness metric together with the mating operation gradually improves the population of hypotheses on linguistic grounds 1), 2) and 3). The mutation operation is designed to introduce diversity into the population. If the algorithm is proceeding towards a sub-optimal solution, a random change of a few parameter values gives the algorithm a chance to 'change course' and hypothesize grammars outside of the current population. The single remaining hypothesis at the end of learning is the grammar the GA found during its search of the parameter space that would best be able to generate all of the sentences of the target language.

Clark's work in computational modeling of parameter setting is noteworthy because it breaks a variety of pre-existing molds. One is that his model is not *error-driven* in the spirit of many pre-parameter setting acquisition algorithms (Wexler & Culicover, 1980) in that the learner may change its hypothesis pool even if one or more of the hypotheses in the pool are able to assign a correct syntactic structure to the input sentence. This offers Clark's model several advantages, it is robust to noise, does not in principle fall prey to subset-superset relationships between languages in the domain, hence does not explicitly need to implement the Subset Principle,[16] and is able to gradually improve the learner's hypotheses not only in the case that the grammars produce many parsing violations (as a strictly error-driven learner would), but also in the case that the grammar pool is correctly parsing the input; not being error-driven allows Clark's model to take advantage of "expected events, not [just] to occurrence of unexpected events" (Kapur, 1994, p. 510).

As mentioned above, it is also the earliest work that exposes a significant quandary for the original concept of triggering as a model for syntactic parameters – parameter setting cannot be a transparent automatic enterprise, it must face and navigate a host of complications due to parameter interaction

[15] The order of the steps differs slightly from Clark's.
[16] Though there is much discussion of a standard variety of the Subset Principle in Clark (1992), the probabilistic version Clark employs, i.e., the folding of SP into the fitness metric is distinctly different.

and parametric ambiguity. Clark's solution is to take a radically different approach than classical triggering – parameter settings are acquired through a search of the hypothesis space of grammars. The search is non-deterministic (i.e., trial-and-error), but is guided (presumably) advantageously by the pool of most fit grammars. The pool is updated from generation to generation randomly after each sentence is encountered; parameter values change from one input sentence to the next input sentence. Clark abandons the deterministic triggering model.[17]

Clark's research marks the beginning of a two-way divergence in approaches to computationally modeling syntactic parameter setting:

[17] Around the same time there was complementary work in the acquisition of generative phonology (Dresher & Kaye, 1990) which posited similar viewpoints concerning parameter interaction and ambiguity. However Clark argued for a radically different parameter setting mechanism than triggering – a non-deterministic, trial-and-error approach, whereas Dresher and Kaye in their learning model tried to maintain much of the determinism associated with the classical concept of triggering.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

1) a *nondeterministic approach* that portrays the learning process as a largely trial-and-error (non-deterministic) search through the space of possible grammars until a grammar is found that licenses all the target sentences encountered by the learner which makes ambiguity-induced errors along the way, and

2) a *deterministic approach* that includes tests to identify and discard ambiguous triggers, so that learning can be based solely on unambiguous information in order to avoid errors. This second abides by what has been dubbed the *Parametric Principle* (Sakas, 2000; Sakas & Fodor, 2001) – correctly set the value of every parameter independently of evaluating whole grammars – which echoes the spirit of classical trigger model.

There are advantages and disadvantages to both approaches. I return to the significance of this 'fork in the road' below and present some conjectures about what directions computational modeling of P&P acquisition will take at the end of this chapter, but for now let's wind up discussion of Clark's non-deterministic approach.

A nontrivial problem with GA models in general, and which Clark also acknowledges is true of his parameter setting model in particular, is what Clark refers to as the *endgame problem*. Basically, as the population (pool of most fit hypotheses) approaches an optimal hypothesis, the diversity in the pool decreases due to the fact that the genetic operators are being applied over more and more similar hypotheses. Linguistically, it may be the case that the GA is considering a small pool of grammars that differ "from the target by only one or two parameter settings that are not represented in the population." (Clark, 1992 p 135) If this situation occurs, the GA's only egress is to rely on random mutation of the genes (parameter settings) in the pool (see footnote **Error! Bookmark not defined.**) which is inefficient and becomes even more inefficient if the expression of the incorrect parameter value occurs in the linguistic environment relatively infrequently.

For example, young children typically encounter few sentences that contain more than one clause (degree-0 sentences consist of exactly one clause, degree-1 sentences consist of two clauses, etc.). It's easy to imagine a situation in which the GA has a small population of grammars that are near optimal in their ability to license degree-0 (single clause) sentences, but that all have one or two parameters governing long distance dependencies misset to the wrong parameter value. As an illustration, consider Chinese as the target language and all the hypotheses in the pool are able to correctly parse degree-0 Chinese sentences but all also have a parameter specifying global anaphoric antecedents set to –Global (Chinese allows global anaphors to be globally bound to their antecedents), i.e., the +Global gene is not to be seen in the population. The GA could introduce the +Global gene into the pool and give it a reasonable chance to survive only in the fortuitous case that: 1) the mutation operation picks the +/-Global parameter to mutate from –Global to +Global, *and* 2) a rare degree-1 sentence which expresses the correct +Global value appears in the input stream before the mutated grammar "dies off", *and* 3) the mutated grammar is chosen as one of the two hypotheses to mate during an iteration the when such a degree-1 sentence occurs. If any of these three events were not to come to pass, the +Global

gene would be doomed to die out of the population, necessitating all three be realized later during the course of acquisition. Thus, the probability is very low that the GA will converge to a single, correct hypothesis within a reasonable amount of time; it does not meet the feasibility criterion. Although this inefficiency is a direct product of the search heuristics employed by the GA, it is greatly exacerbated by the well-documented scarcity of parametric expression of certain syntactic phenomena in the input encountered by children (see discussion in Chapter XXX on the Argument from the Poverty of the Stimulus).

Another concern not raised explicitly by Clark but by others (Nyberg, 1992; Yang, 2002) is the use of multiple grammars to parallel parse a single input sentence. Clearly human adults do not engage in massive parallel parsing of sentences (although it may be that there is some limited parallel parsing employed (Gibson, 1991)). It's not specified in Clark's articles how many grammar hypotheses should be maintained in the population as learning proceeds but it seems likely, based on other successful applications of GAs in other domains, that it would be considerably more than two or three – especially at the outset of learning – which undermines the psychological plausibility of the model (cf. Daelemans, 2002).

Finally the fitness metric itself has been brought into question (Dresher, 1999) on grounds that knowledge of subset-superset relations should arise from intensional or I-language *cues*, rather than a list provided by UG as Clark argues. And, more interestingly, that the heuristic of preferring grammars that return fewer parameter violations is not a useful strategy in the domain of human languages. Dresher gives an example from generative phonology where if there were a change in just one parameter "every syllable [of the target language] will receive the wrong stress." Dresher continues:

> 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. (Dresher, 1999, p. 16)

Dresher is referring to a domain's *smoothness*. A smooth domain is one in which there is a tight correspondence between changes in the grammar space and changes in the languages the grammars generate.[18] Dresher is maintaining that a parameterized domain of metrical stress is *un*smooth and hence Clark's implementation of fitness is untenable for stress. A similar argument can be made for the domain of word-order parameters.

For example, consider a P&P grammar, *G*, which consists of a vector of parameter values including the value $P_i(v)$ and that *G* licenses language *L*. If the domain is smooth then changing $P_i(v)$ to $P_i(v')$ should

[18] The term *smoothness* as used here is a description of the language/grammar domain. It is sometimes also used as a description of the learning process meaning "gradualness" – when the learner considers successive grammar hypotheses that differ minimally from one another.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

produce a grammar that licenses a language very similar to *L*. This is probably more true of some parameters than others. If we take *G* to be the parameter settings for English and $P_i(v)$ to be –pied-piping (English allows prepositional heads to be 'stranded' away from their objects, see above). If –pied-piping is changed to +pied-piping with all other parameters remaining the same we would have basically a schoolmarm-approved version of English where sentences are not allowed to end with a preposition and nothing else in the English language is affected. However if we were to change a single parameter value that would flip English into a head-final language from a head-initial language, the new language would share very few sentences with English. We might arrive 'back' to English by changing the verb movement parameters to help place the verbs into more English-like positions in the generated sentences. However in doing so we would be changing more parameters away from their correct ones – moving the *grammar* further from the correct parameter values of English – and at the same time moving the hypothesized *language* closer to the language of English. This is analogous to the case Dresher described for metrical stress above.

To the best of my knowledge there are no formal definitions of smoothness, but the concept of a smooth domain is generally accepted to mean that *all* grammar variation behaves like the pied-piping parameter. Dresher's point is that Clark's fitness metric would work only in the case that the domain of human languages is smooth; a point that has been strongly argued against by Chomsky and others. For example, Chomsky writes:

> There is no simple relation between the value selected for a parameter and the consequence of this choice as it works its way through the intricate system of universal grammar. It may turn out that the change of a few parameters, or even of one, yields a language that seems to be quite different in character from the original. [Such as the headedness parameter described above.] (Chomsky, 1988, p. 63)

We return to discussion of the relationship between domain smoothness and non-deterministic learning below.

In summary, Clark's groundbreaking work unveiled severe limitations of classical triggering theory. Under the umbrella assumption that human language is rife with ambiguity and that grammars contain interactions among syntactic parameters that are difficult to tease apart Clark set the stage for the computational modeling of P&P acquisition for over two decades. Many have adopted Clark's observation that linguists' view of automatic triggering is too weak to cope with ambiguity – acquisition requires a computationally motivated strategy to search over the space of grammars.

2.2 Gibson and Wexler: Triggering as Search

In 1994, Edward Gibson and Kenneth Wexler (henceforth G&W) in a seminal article (Gibson & Wexler, 1994) present the Triggering Learning Algorithm (TLA). The algorithm is appealingly simple. The learner tests its current grammar hypothesis on an input sentence. If the hypothesis licenses the input, the

learner does nothing. If the hypothesis does not license the input, then the learner randomly flips a single parameter. If the new hypothesis licenses the input, then the new hypothesis is adopted. Otherwise the learner retains the original hypothesis. As G&W note, the learner is constrained in three ways. First, following Wexler and Culicover (1980), the learner is *error-driven*; it will only change its hypothesis if the current hypothesis generates an error when analyzing an input. In an error-driven learning system, convergence falls out naturally from the learning algorithm; once the target has been attained the algorithm canonically will never change its (correct) hypothesis (assuming there's no noise in the input); there is no need for a stopping criterion or an oracle that has knowledge that the target hypothesis has been achieved. Second, following Clark (1992) the learner employs the *Single Value Constraint* (SVC) (see discussion above). Finally, the learner is constrained to only adopt a new hypotheses in the case that the new hypothesis can license the current input sentence when the learner's current hypothesis cannot, i.e., when the new hypothesis is 'better' than the current one. G&W name this the *Greediness Constraint*.

Given:

- n, the number of parameters
- $G_{curr}=(p_1, p_2, \ldots,p_n)$, the current grammar (i.e., a vector of parameter values), where each p_i is a either a + value or a – value
- G_{new}, like G_{curr}, but only hypothesized temporarily, for a single sentence

a sketch of the algorithm follows.

Gibson and Wexler's TLA

- *For each sentence s:*
  - Randomly set all p_i's in G_{curr}
  - If G_{curr} can parse s, then do nothing (Error-driven)
    otherwise:
    - Set all the p_i's in G_{new} equal to the p_i's in G_{curr}
    - Pick a random j, where $1 \leq j \leq n$
    - Toggle the value of p_j in G_{new} from either: a + to a –,or, a – to a + (SVC)
    - Parse s with G_{new}
    - If G_{new} can parse s, the G_{curr} becomes G_{new}, otherwise do nothing (Greediness)

There are several attractive aspects of G&W's work. Greediness, the SVC and being error-driven together ensure conservatism. Like Clark's GA model, learning is gradual; the constraints prohibit the learner from jumping to radically different grammars on every input. More importantly, G&W restrict the computational resources required by the learner. The TLA requires no memory, and at most the application of only two grammars on a single input. These are significant improvements on the GA model where multiple grammars are tested simultaneously on the input and the GA's pool of most fit hypotheses implement a form of memory.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

Another noteworthy aspect is the language domain that G&W construct to test the TLA. G&W were the first to construct a space based on linguistically authentic parameters. They incorporated three parameters into their domain generating $2^3 = 8$ abstract but linguistically motivated languages, and G&W use it to show under what circumstances the domain is learnable by the TLA. This methodology is still actively pursued today, though its scale has increased.

Finally G&W were the first to make the distinction between *global* and *local triggers*. A global trigger for parameter P_i is a sentence that requires P_i be set to a specific value v, denoted as $P_i(v)$. If P_i is set to another value v' then the sentence cannot be parsed by the grammar – regardless of any of the other parameter values. A global trigger must also be a sentence of all languages generated by grammars with $P_i(v)$. Like a global trigger, a local trigger for parameter P_i is a sentence that requires P_i be set to a specific value v but only when the other parameters are set to specific values. If the other parameters are set differently, then $P_i(v)$ would work just as well as $P_i(v')$, i.e., the sentence would not be a trigger for $P_i(v)$. Local triggers exist only in some languages that require $P_i(v)$, those that have the other parameters set to specific values. A useful extension of global and local triggers is presented in the section on Sakas and Fodor below. Interestingly the G&W domain contains only local triggers (and no subset-superset relationships among the languages, so the TLA does not need to implement the Subset Principle).

Given that the TLA model embodies some psychologically and linguistically desirable features, and that finite parameter spaces are presupposed to be easy to learn, a somewhat surprising result is that the TLA fails to acquire some,[19] but not all eight of the target languages grammar hypothesis within this very simple parametric framework. This is because the learner gets 'trapped' forever in an incorrect hypothesis, or *local maxima* on the way to the target grammar due to a lack of local triggers generated by the target grammar. Basically if the learner hypotheses one of these local maxima grammars there are no sentences that will allow the TLA to make a move towards the target grammar. Though G&W explore a number of solutions to this problem including defaults and parameter ordering or maturation. (cf., Bertolo, 1995b), later research (Berwick & Niyogi, 1996; Niyogi & Berwick, 1996) which elegantly formulates the learning process as a Markov chain establishes that there are inevitable situations in which the TLA is doomed to fall prey to local maxima in G&W's three parameter domain; i.e., fails to meet the learnability criterion.[20] This is supported by two simulation studies on two larger domains in which the TLA failed to acquire the target grammar a large percentage of the time (Kohl, 1999; Fodor & Sakas, 2004).

Even if the domain of human languages does indeed supply local triggers for all parameters, results from probabilistic analysis of the TLA (Sakas, 2000; Sakas & Fodor, 2001) suggest that given a uniform

[19] The number depends on which grammar is the target and which is the starting hypothesis grammar.

[20] Two interesting studies, taken together, point to the fact that local maxima are caused by both the formulation of the domain and the learning algorithm applied to the domain. Turkel (1996) demonstrates that a genetic algorithm does not encounter any local maxima when acquiring the grammars of G&W's domain. Frank and Kapur (1996) show the number of local maxima encountered by the TLA is reduced if three different (but linguistically sensible) parameters are used to generate the languages of G&W's domain

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

distribution of ambiguity as the domain is scaled up to incorporate a reasonable number of parameters the number of sentences required for TLA to converge increases exponentially – over a billion sentences would be required to acquire a target grammar of 30 parameters. This dovetails with results from Niyogi and Berwick (op cit.) which shows that the TLA does worse than a random step learner which chooses grammars randomly without any of G&W's three constraints, and is also supported by Fodor and Sakas' (op cit.) simulation study. Hence, the TLA also fails to meet the feasibility criterion.

However, Sakas (2000) also provides results demonstrating that the TLA does succeed as a feasible model of acquisition in the case that the language domain being acquired is smooth (and assuming there are no local maxima in the domain). This is a recurring theme in the current chapter (see discussion of Clark above, and Yang below). I propose the following hypothesis:

(H1) Nondeterministic computational search heuristics work well in smooth P&P syntax domains.

We return to this in the section on Yang. For now we adopt the premise that the domain of human syntax is not smooth (see Section 2.1, and that the TLA is ultimately not viable as an account of human language learning though G&W's work is noteworthy for its methodology as the first computational attempt at modeling syntax acquisition with a minimal amount of computational resources.

2.3 Fodor: Unambiguous Triggers

In response to G&W's work on the TLA, Janet Dean Fodor introduced a family of learners that does abide by the Parametric Principle *The Structural Triggers Learner* (*STL*) (Fodor, 1998a, b; Sakas & Fodor, 2001) that takes a significantly different view of the triggering process. Rather than guessing a grammar hypothesis or hypotheses to adopt, as both Clark's and G&W's models did, Fodor's STL makes uses of structural information generated by the parser. The key to how the STL operates is that parameter values are not simply +'s or –'s but rather bits of tree structure or *treelets*.

Each *treelet* contains (the minimal) structural information required to define a point of variation between languages, i.e., a treelet *is* a parameter value. For example, in the CoLAG domain described below, the parameter value that licenses Wh-Movement is a treelet consisting of a Spec,CP with a [+WH] feature. UG demands that a Spec,CP[+WH] dominates a morphologically wh-marked lexical item and is associated with its trace somewhere else in the sentence. If a grammar contains this treelet, then wh-marked items are fronted under Spec, CP[+WH].

In this picture, triggers and parameter values are ingredients of both grammars *and* ingredients of trees. Suitable as ingredients of grammars, they are all combined into one large grammar (termed a *supergrammar*) which the parser applies to the input in exactly the same way as any other grammar is applied. Parameter values (suitable as ingredients of trees) are present in the parse trees output by the parser so that the learning device is able to see which of them had contributed to parsing an input sentence and would know which to adopt. UG provides a pool of these schematic *treelets*, one for each parameter value, and each natural language employs some of them. Used as a trigger by the learning mechanism, a treelet is detected in the structure of input sentences. As a parameter value, it is then adopted into the learner's current grammar and is available for licensing new sentences.

Thus, a grammar for the STL is a vector of treelets rather than a vector of +'s and -'s. The STL is error-driven. If the current grammar, G_{curr}, cannot license *s*, new treelets will be utilized to achieve a successful

parse.[21] Treelets are applied in the same way as any "normal" grammar rule, so no unusual parsing activity is necessary. The STL hypothesizes grammars by adding parameter value treelets to G_{curr} when they contribute to a successful parse.

The basic algorithm for all STL variants is:

1. If G_{curr} can parse the current input sentence, *s*, retain the parametric treelets that make up G_{curr}.
2. Otherwise, parse the sentence making use of any parametric treelets available in the supergrammar, giving priority to those in G_{curr}, and adopt those treelets that contribute to a successful parse. We call this *parametric decoding*.

Because the STL can decode inputs into their parametric signatures, it stands apart from other acquisition models in that it can detect when an input sentence is parametrically ambiguous. During a parse of *s*, if more than one treelet could be used by the parser (i.e., a *choice point* is encountered), then *s* is parametrically ambiguous.[22] Note that the TLA does not have this capacity because it relies only on a can-parse/can't-parse outcome and does not have access to the on-line operations of the parser. Originally, the ability to detect ambiguity was employed in two variations of the STL: the *strong STL* and the *weak STL*.

The strong STL executes a full parallel parse of each input sentence and adopts only those treelets (parameter values) that are present in all the generated parse trees. This would seem to make the strong STL an extremely powerful, albeit psychologically implausible, learner.[23] However, it is not guaranteed to converge on the target grammar (G_{targ}). The strong STL needs *some* unambiguity to be present in the structures derived from the sentences of the target language. For example, there may not be a single input generated by G_{targ} that when parsed, even in parallel, yields an unambiguous treelet for a particular parameter.

Unlike the strong STL, the weak STL executes a psychologically plausible left-to-right serial (deterministic) parse.[24] One variant of the weak STL, the *waiting STL*, deals with ambiguous inputs abiding by the heuristic: *Don't learn from sentences that contain a choice point.* These sentences are simply discarded for the purposes of learning. This is not to imply that children do not parse ambiguous sentences they hear, but only that they set no parameters if the current evidence is ambiguous. It is important to note that the strong STL and the waiting STL do not perform a search over the space of possible grammars; they decisively set parameters. This is quite unlike either Clark's GA model, or

[21] In addition to the parameter treelets, UG principles are also available for parsing, as they are in the other models discussed in the chapter.
[22] The account given here is idealized. In fact, the picture is complicated by the existence of within-language ambiguity (*One morning I shot an elephant in my pajamas*) and the fact that a temporary choice point might be disambiguated by the end of an input sentence (see discussion in Fodor, 1998b).
[23] It is important to note that the strong STL is not a psychologically plausible model. Rather, it is intended to demonstrate the potential power of parametric decoding (Fodor, 1998a; Sakas & Fodor, 2001).
[24] With the capability of reanalysis when faced with garden path sentences (Fodor, 1998b).

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

G&W's TLA model, which do engage in grammar space search. Thus the STL model is closer to the original concept of triggering.

As with the TLA, these STL variants have been studied from a mathematical perspective (Bertolo, Broihier, Gibson, & Wexler, 1997a, b; Sakas, 2000; Sakas & Fodor, 2001). Bertolo et al. offer a formal learnability proof that the STL will fail in principle in domains that exhibit certain patterns of overlap between languages due to an insufficiency of unambiguous triggers. However, the proof relies in part on the unsupported premise that a deterministic earner cannot employ default parameter values. Probabilistic analysis conducted by Sakas, and Sakas and Fodor point to the fact that the strong and weak STLs are extremely efficient learners in conducive domains with some unambiguous inputs but may become paralyzed in domains with high degrees of ambiguity.[25] These results among other considerations spurred a new class of weak STL variants which we informally call the *guessing STL* family. (Fodor, 1998b; Fodor & Sakas, 2004) The basic idea behind the guessing STL models is that there is some information available even in sentences that are ambiguous, and decoding can exploit that information.

In its simplest form a guessing STL is a waiting STL that doesn't discard any inputs. When a choice point is encountered during a parse, the guessing STL model adopts a parameter treelet based on any one of a number of heuristics, e.g., randomly, favor treelets without traces or that abide by other universal parsing strategies, favor treelets that have worked well in the past,[26] etc. Strictly speaking, the guessing STL violates the Parametric Principle and does perform a nondeterministic search of the hypothesis space constrained by one its heuristics. However, simulation studies on an abstract domain[27] consisting of 13 linguistically plausible word-order parameters (Sakas, 2003; Fodor & Sakas, 2004) show that the guessing STL variants perform significantly more efficiently than the TLA (when no local maxima were encountered) – approximately a 10-fold improvement.[28] This is because the guessing STL still makes extensive use of decoding and certain parameters are easy to set due to unambiguous information.

Preliminary results indicate that some parameters value never change after a few initial sentences, even though the guessing STL is 'allowed' to change them, while others will change during the acquisition process. For example once the pied-piping parameter is set, the learner will never have the *need* to change it back – it will never encounter a sentence (other than noise) that requires the opposite value of

[25] They used a mathematical formulation of the domain in which language membership of an input was determined by a probability; languages were not collections of strings. See Yang (2002) for a similar approach to domain modeling.

[26] The first two of these have been implemented (Fodor & Sakas, 2004), the third has not. Discussion can be found in Fodor (Fodor, 1998b). Though not identical, Fodor's notion is similar to a later proposal by Yang which has been implemented. See discussion below in Section 2.4.

[27] The domain was constructed much in the spirit of G&W's domain, though it's much larger (3,000+ languages as opposed to G&W's 8) and covers many more syntactic phenomena (13 parameters as opposed to G&W's three). We return to discussion this domain in the section on Sakas and Fodor.

[28] The TLA converged on the target grammar 12% of the time. The waiting and strong STL variants acquired the target grammar only 25% and 26% of the time respectively. The guessing STL variants always converged on the target grammar. Note that none of the learners were faced with subset-superset choices in this study.

the parameter. This is unlike the verb movement parameters and the obligatory topic parameter described above in which parameters interact, so the guessing STL is forced to employ one of its heuristics to search the space of these parameters until it settles on the correct values. The end result is that the Parametric Principle is *effectively* being adhered to – at least for some parameters. Why does this help? Because abiding by the Parametric Principle cuts the search space in half every time a parameter is set. Consider a space of 10 parameters which specify 2^{10} = 1,024 grammars. If a single parameter is set permanently, the learner need only consider the other 9 parameters which reduces the search space to 2^9 = 512 grammars. If 4 out of 10 parameters are fixed, the search space is reduced to 2^6 = 64 grammars; the size of the search space is reduced exponentially as more and more parameters are set. The TLA never reduces the search space since it relies on only parse/can't parse information from the parser; i.e., it does not perform decoding. Hence, the search space is never reduced. The guessing STL models only needs to search the space of 'problematic' interacting parameters which makes it far more efficient than the TLA.

In summary, Fodor's STL model is a departure from previous computational models that search the space of all possible grammars. The STL uses the parser to decode sentences into bits of structure – treelets - that embody parameter values. The STL is then able to choose from among the treelets that result from decoding and incorporate them into the current grammar hypothesis. The purest of the STL variants (waiting and strong) are very close to the classical triggering model of parameter setting, though not as 'automatic' since they require effort on the part of the parser (still, as Fodor argues, parsing is needed for comprehension anyway). However, they clearly suffer from a lack of unambiguous inputs. Results from simulation studies of the guessing STL variants, although further from the classic triggering model since they do search a partial space of hypotheses, indicate that the STL is a viable candidate as a model of human language acquisition.

2.4 Yang: Variational Learning

Charles Yang in a well-received book (Yang, 2002) puts forward the argument that it is necessary for any learner to perform well in domains without unambiguous inputs since the "general existence of unambiguous evidence has been questioned (Clark, 1992; Clark & Roberts, 1993)." He provides an elegant statistical parameter setting algorithm which is indeed capable of converging to a target grammar in a domain that contains no unambiguous evidence (Straus, 2008).

The algorithm maintains a vector of weights, each weight in the vector is associated with a parameter and represents the probability that a particular value for that parameter is hypothesized after encountering an input sentence. If the probability is above 0.5, then the + value will be more likely hypothesized by the learner, if it's under 0.5 then the – value will more likely be hypothesized. The weights are 'updated' by the learner based on the outcome of a can parse/can't parse attempt at parsing the current input sentence, *s*. Through the process of updating, Yang's weights serve as a form of memory of how well particular parameter values have performed on past inputs. Note that weights are a well-formulated construct akin to Fodor's notion of activation levels of parametric treelets (Fodor, 1998b, p. 360).

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

Given:

- n, the number of parameters
- a vector of weights, $W=(w_1, w_2, \ldots, w_n)$, where each w_i is stipulated to be between 0 and 1
- the current grammar (i.e., a vector of parameter values), $G_{curr}=(p_1, p_2, \ldots, p_n)$, where each p_i is a either a + value or a – value

a sketch of the algorithm follows.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

Yang's Variational Learner

- For each w_i in W, set w_i to 0.5
- For each input sentence s:
    - For each parameter p_i in G_{curr}:
        a. with probability w_i , choose the value of p_i to be +;
        b. with probability 1 - w_i , choose the value of p_i to be – ;
    - Parse s with G_{curr}
    - Update the weights in W accordingly.

Yang gives two methods of updating the weights. The details are not important to the discussion here, but the basic idea is to 'reward' G_{curr} if the parse of s with G_{curr} is successful (or a number of parses on different inputs are successful), and 'punish' the weights if the parse (or parses) fail. Rewarding or punishing inches the weights in one direction or the other depending on the current parameter values that are in G_{curr}.

Updating by rewarding:

If a parameter in G_{curr} is +, then the weights are nudged towards 1.0. If a parameter in G_{curr} is –, then the weights are nudged towards 0.0.

Updating by punishing:

If a parameter in G_{curr} is +, then the weights are nudged towards 0.0. If a parameter in G_{curr} is –, then the weights are nudged towards 1.0.

For example if G_{curr} is rewarded (i.e., G_{curr} *can* parse the current input) and parameter p_i in G_{curr} is set to +, and its weight, w_i, is equal to 0.6, then 0.6 is increased by some amount. Rewarding has the effect of the learner being more likely to hypothesize a + value in the future (step (α) in the algorithm). If G_{curr} is punished however, 0.6 is decreased by some amount, hence a + value is less likely to be hypothesized in the future. The same holds for – values. The amount to increase or decrease the weights is determined by the Linear reward-penalty ($L_{R\text{-}P}$) scheme (Bush & Mosteller, 1955; See Pearl, 2007, who proposes an interesting Bayesian alternative that offers some advantages). The picture to paint is one of coexisting grammars that are in competition with each other in the attempt to match the observed linguistic evidence.

There is much of interest in Yang's algorithm as compared to previous computational models. First, like Clark's GA model, it is not error-driven. Unlike an error-driven learner which waits until there is a misfit between the current hypothesis and the input data, a Variational learner through rewarding and punishing makes progress both when the data fits the current grammar, and when it doesn't. Secondly, it is resource-light. All that is needed in terms of memory (in addition to the parameter values themselves) is a number of weights equal in number to the number of parameters. This is a significant improvement over Clark's GA model which requires a pool of grammar hypotheses. Another

improvement over Clark's GA is the relatively simple update mechanism of the weights on single parses. Clark's model requires multiple parses over batches of sentences (though cf., Daelemans, 2002) . Thirdly, although the resources required of Yang's learner are somewhat more than what is required of G&W's TLA, it is in principle impervious to pitfalls of local maxima since there is always a probability greater than 0 that the learner will temporarily jump to a different area of the parameter space, a feat impossible for the TLA to perform. Finally, a recent dissertation (Straus, 2008) proves that at least one method of Yang's method updating the weights ("The Naïve Parameter Learning" model) is guaranteed to converge in most any P&P domain[29] - including domains containing only ambiguous data. Hence, unlike Fodor's STL, Yang's learner need not rely on the existence of unambiguous input.

In its own right, Variational Learning makes two significant contributions to computational modeling of parameter setting. Walter Daelemans, in a review of Yang's dissertation (which serves as the basis for Yang's book) puts it well:[30]

> In conclusion, I think Charles Yang's dissertation is an important milestone in P&P-based theories of language learning, and Variational Learning deserves to be widely studied. For the first time, a general learning method is combined with a UG-based hypothesis space into an acquisition model that seems to have largely the right formal characteristics and that improves upon earlier proposals. Especially interesting is the fact that the approach provides a new tool for the study of both child language development and language change in an objective, corpus-based, and quantitative way. (Daelemans, 2002)

To unpack Daelemans' comments: by *general learning* Daelemans is making reference to what is often called *domain general learning* – learning methods that can be applied to any domain without resorting to prior knowledge of, or being biased by specific facts of the domain that is being acquired. This can be contrasted readily with triggering theory and Clark's approach, each require that learner have either knowledge (of what's serves as a trigger), or bias (favor subset languages) specific to language. Yang's learner requires no specific knowledge or bias (the parse test is presumably not part of the learner, only the ability of the learner to observe the result). The point is, that in principle, Variational learning is not specific to language learning. Hence, Yang's work marks the first time that domain general learning is employed in a "UG-based" model of language acquisition.[31] This is significant because historically

---

[29] The proof requires that all the grammars, other than the target, are less likely than the target grammar to parse the sentences of the target language. This implies that domain does not contain a language which is a superset of the target. If it did, then the probability of parsing an input sentence would be the same for both the superset grammar, and the target grammar. The same would be true of a language that was *weakly equivalent* to the target language – exactly the same sentences of the target language, licensed by the target grammar and (at least one) other grammar. Domains with these characteristics are not covered by the results in the dissertation.

[30] In other parts of the review, Daelemans is somewhat critical of Yang's quick dismissal of "related approaches in (statistical) machine learning, computational linguistics and genetic algorithms."

[31] This might be taken as a little strong. Clark's genetic algorithm follows a standard implementation used by many AI researchers working in a wide range of domains. The TLA could also be used to search a non-linguistic domain (assuming that the domain can be meaningfully parameterized). Daelemans' point is that Clark's fitness metric, and G&W's constraints on learning were both motivated by linguistic considerations, presumably giving the TLA a *domain specific language learning* bias. Yang's learner, on the other hand, employs a generic, domain-neutral heuristic: reward hypotheses that cover the current input datum, punish those that don't.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

language acquisition research in the generative tradition has taken the stance that the learning mechanism must be language specific do to the paucity of input that children receive (see discussion in Chapters XXX and XXXX in this volume).

The second point Daelemans makes, and the more important one, is that Yang's research includes explication of how Variational Learning can explain developmental data – specifically of Null Subject data produced by children learning English – and language change data. Language change aside, to date it's the only study of how a computational model of syntactic parameter setting is shown to perform in accordance with actual data from child speech.

Despite the impact of Yang's work, there are two fronts on which weaknesses can be identified. The first one, ironically, concerns one of the abovementioned strengths of Yang's work – the ability of the Variational Learner to make predictions about developmental data. Snyder (op cit.) and Sugisaki and Snyder (2005) argue that "the acquisition of preposition-stranding in English poses a serious challenge" to Yang's learner, because if the + and – values of the pied piping parameter were in competition there would be observable effects of pied-piping in children's utterances. They present data from a previous study (Sugisaki & Snyder, 2003) which targets wh-questions. They studied English corpora from 10 children in the CHILDES database (MacWhinney, 2000) and found that none of the children produced a wh-question with pied-piping – the preposition was consistently stranded. This, they argue, is not consistent with the prediction Variational learning would make. Until the – value (as in English) of the pied piping parameter was rewarded sufficiently, the Variational Learner would have a non-zero chance of hypothesizing the + value. Indeed at the beginning of learning there is an equal chance of choosing either value.[32] Unlike the pied-piping parameter which produces observable effects in children's utterances, the Yang's Null Subject data is the result of modeling grammar competition where English speaking children are making errors of *omission*, omitting required subjects, so there are no observable effects in the utterances of the children when entertaining an incorrect parameter value. In general, any nondeterministic learning model would predict that there would errors of *comission* in developmental data – grammatical errors that are observable in children's utterances. Snyder (op cit.) argues extensively that this is not the case and that developmental data is consistent with a deterministic classical triggering model of parameter setting.

Another potential weakness of Variational learning involves the feasibility of Yang's model in terms of number of inputs required to converge. Yang states that "about 600,000 sentences were needed for converging on ten interacting parameters." Preliminary results from this author's simulation studies comparing Fodor's STL models, the TLA and Variational learning on the CoLAG domain of 13 parameters (see below) concur: The Variational Learner (and the TLA) requires an order of magnitude more input sentences to converge than the guessing STL. This in and of itself does not argue conclusively that Yang's model is not viable as a model of human language acquisition (perhaps the STL models converge *too* quickly!). However, Yang's result (of 600,000 sentences) was generated by a simulation study of an artificial domain in which a high level of smoothness was imposed – grammars that shared more parameter values with the target grammar (effectively) generated languages[33] more similar to the target

[32] They also argue that the time course of development with respect to wh-questions and preposition stranding is incorrectly predicted by Yang's learner.

[33] Yang used a probabilistic framework to craft the language domain similar to that of Sakas (2000) and Sakas and Fodor (2001). See footnote 25.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

language than grammars with fewer parameter values in common with the target (see discussion above). Given a smooth domain Yang's reward/punish heuristics will accelerate the Variational Learner towards the target at an increasing rate. That is, as learning proceeds the learner will be more and more likely to hypothesize grammars that are more and more similar to the target grammar, i.e., that have more and more parameters in common with the target. Recall that mathematical modeling of the TLA showed a feasible level of performance in a smooth domain, but not in other domains (Sakas, 2000). It remains an open question to what extent the Variational Learner can cope with unsmooth parametric domains.

Despite these two weaknesses (the latter, only a potential weakness) Yang's Variational Learner stands as the state-of-the-art in non-triggering, nondeterministic computational modeling of parameter setting.

2.5 Sakas and Fodor: Trigger Disambiguation.

Since Clark's identification of the problems of parametric ambiguity and parameter interaction, computational modeling of parameter setting mostly employ non-deterministic search strategies over the parameter space. Recent work by Sakas and Fodor (Submitted) suggests that the problem of parametric ambiguity might be overcome within a deterministic framework.

This work makes use of a domain containing 3,072 natural-language-like artificial languages used for testing models of syntactic parameter setting: The CUNY-CoLAG domain[34] (or CoLAG for short). All languages in the domain share general structural principles, which constitute the Universal Grammar (UG) of CoLAG. The individual languages are generated by grammars defined by 13 binary syntactic parameters which control familiar phenomena such as head direction, null subjects, wh-movement, topicalization, and so forth.

In the spirit of previous work (Gibson & Wexler, 1994; Bertolo, et al., 1997a, b; Kohl, 1999), the CoLAG domain has a universal lexicon (e.g., *S* for subject, *O1* for direct object, *O2* for indirect object, *-wa* for topic marker, etc.); sentences (or more accurately sentence patterns) consist of sequences of the non-null lexical items, e.g., *S-wa O1 Verb Aux* or *O1-wh Verb S O2*. Grammars are non-recursive and the domain contains 48,086 sentences (on average 827 per language), and 87,088 fully specified syntactic trees which means there is substantial syntactic ambiguity (another indication of ambiguity is that every sentence is licensed by 53 grammars on average).[35]

Using computer database queries, Sakas and Fodor (hereafter, S&F) discovered that almost half of the parameter values lack unambiguous triggers in some or all of the languages which need those parameter values. For three of the 13 parameters there are insufficient unambiguous triggers for *both* (+ and -) values. However, by making use of some standard default values (e.g., a value that licenses subset languages), and using a non-standard "modest toolkit" of disambiguating strategies: *between-*

[34] Spelled out: City University of New York – Computational Language Acquisition Group domain.
[35] The trees are constructed in the manner of Generalized Phrase Structure Grammar using SLASH categories to create movement chains.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

parameter-defaults, and *conditioned triggers* (described below) they were able to establish unambiguous triggers for all non-default parameter values in all of the CoLAG languages. S&F's study suggests that contra previous work, a computational model of parameter setting could proceed deterministically by setting each parameter only after encountering an unambiguous trigger for the required value (ignoring ambiguous input). Since unambiguous triggers are reliable, each parameter would need only be set once without the need to engage in trial-and-error exploration of the parameter space (see discussion of the *Parametric Principle* above).

S&F emphasize the distinction between *E-triggers* and *I-triggers* (cf., Lightfoot, 1999; Lightfoot, 2006). An E-trigger is an observable property of a sentence, and an I-trigger is an innate piece of grammatical structure that realizes the observable E-trigger. Consider again the pied-piping parameter which specifies whether or not prepositional objects must always reside in a prepositional phrase. The E-trigger would be "non-adjacency of P and O3" whereas the I-trigger would be whatever aspects of the grammar that licenses the O3 being moved out of the PP.[36] I-triggers are exactly the 'treelets' of previous work on STLs (see above) and are available to the child learner as prescribed by UG. Contra E-triggers, I-triggers are not readily observable in the linguistic environment. Though there is significant discussion of the relationship between I-triggers and the E-triggers generate, the results most relevant here revolve around establishing unambiguous E-triggers for every parameter in the CoLAG domain.

S&F also extend G&W's distinction between local and global (E-)triggers by incorporating the notions of *valid* triggers and *available* triggers.[37] A valid trigger is a trigger that can be used (safely) by the learner to set a parameter to a particular value $P_i(v)$. A g*lobally valid trigger* for $P_i(v)$ is what I have been have been referring to as an *unambiguous trigger* here – a trigger that exists only in languages whose grammars have $P_i(v)$ and does not exist in any grammar having $P_i(v')$. A *locally valid* trigger is a trigger that can be used to set $P_i(v)$ given that the settings of other parameters are established. "Availability" specifies the extent to which a trigger exists in the languages of the domain. A *globally available* trigger exists in **all** languages whose grammar has $P_i(v)$ whereas a *locally available* trigger exists in only a subset of languages whose grammars have $P_i(v)$.[38] Note that a globally valid trigger may not be globally available – a distinction G&W do not make.

S&F first searched CoLAG for globally valid E-triggers and found that for 5 (out of the 13) parameters every language generated by either the + or – value contained at least one. For another 5 parameters there existed at least one globally valid trigger for the + value. Thus, for these 5 parameters, by designating the – value as the default, the + value could be reliably triggered in languages requiring the

[36] From S&F: " 'SLASH O3' as in the node label 'PP[SLASH O3]' which by definition dominates a PP whose internal object (O3) has been extracted", though of course the formulation of I-triggers would differ from one theoretical framework to another.

[37] The following presentation of S&F's definitions slightly simplified. However, the specifics are not relevant to the discussion here.

[38] A global trigger for G&W, is by these definitions: a globally available, globally valid trigger. A local trigger for G&W, is a locally valid, locally available trigger.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

+ value, and a learner would never be falsely led to the + value in languages requiring the – (default) value since by definition the globally valid triggers for the + value do not exist in languages that require the – value.

For the remaining three parameters (ItoC Movement, Q-Inversion and Optional Topic) there were languages in CoLAG that lacked globally valid triggers for both the + and – values. S&F make the move to "create" unambiguous triggers by employing two disambiguating strategies; Between-parameter defaults and conditioned triggers.[39] From S&F:

***A between-parameter default*:** A sentence property *s* that is compatible with the marked values of two different parameters P_i and P_j is *stipulated* as triggering P_i, leaving P_j to be set by other triggers. This is helpful in cases where P_i suffers from a shortage of triggers but P_j does not; in effect, P_j makes a *gift* of trigger *s* to P_i.

> **Example:** Absence of an overt topic, which is compatible with both +OptTop and +NullTop, is designated as triggering +OptTop only; where appropriate, +NullTop is set by a different trigger (absence of an obligatory sentence constituent).

***A conditioned trigger*:** A sentence property *s* compatible with two (or more) parameter values v_i and v_j (values of the same parameter or different ones) becomes unambiguous as a trigger for v_i once the value v_k of some other parameters are established. (This is a locally valid trigger in terms of the definitions above.)

> **Example:** the surface word order AuxVOS is compatible with either −ItoC or +ItoC but becomes an unambiguous trigger for −ItoC once the Headedness in CP parameter has been set to Complementizer-final.

S&F take both strategies to be innately endowed. They also point out neither strategy could be employed in an ad hoc fashion; both require that "safety requirements" be met (e.g., do not rely on a default value for v_k when creating a conditioned trigger). Whenever needed in CoLAG, the safety requirements for both strategies were met.

After applying these disambiguation techniques, the remaining 3 of the CoLAG parameters could be set; in the end, all 13 parameters in CoLAG could be reliably set. The key concept is that if unambiguous (globally valid) triggers don't exist in the input, they can be *created* by strategies that a learner could employ. In fact this is precisely what standard (within-parameter) defaults do. Given input that is ambiguous between two values of a parameter, disambiguate by favoring the default value.

Of course, there is no guarantee that this positive result for the CoLAG domain will extend to the full domain of natural languages. Even if it does, the finding that deterministic parameter setting is feasible

[39] Similar strategies have been used in the past for modeling the setting of phonological parameters (Dresher & Kaye, 1990; Dresher, 1999).

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

does not entail that it is the means by which children acquire their grammars; in fact there are several drawbacks of strictly deterministic learning as discussed in the next section. S&F's work, however, re-opens the possibility for future computational (and psycholinguistic) investigation to incorporate components of classical triggering theory.

3 Back to the Future: A return to Classical Triggering Theory?

Sakas and Fodor's work suggests that the turn to non-deterministic, heuristic search in most existing computational models of parameter setting – a radical departure from the original conception of triggering – may have been premature. However, even if the domain of human languages contains sufficient unambiguous E-triggers given a workable model of trigger disambiguation, other linguistic, computational and psycholinguistic issues concerning strictly deterministic parameter setting have been raised in the literature.

- Deterministic learning is not robust in the face of noisy input – input that contains linguistic phenomena not in the target language. A deterministic learner runs the risk of mis-setting a parameter based on noise with no recourse (by definition) to re-setting the parameter to the correct value (Nyberg, 1992; Kapur, 1994; Fodor, 1998b).

- Determinism also entails immediate shifts from one grammar to another rather than a gradual change over time (van Kampen, 1997; Yang, 2002).

- A deterministic learning mechanism that is 'perfect' and only sets parameters correctly precludes an explanation of the development of creole languages based on pidgin language input (and language change in general) that engages acquisition at its center (Lightfoot, 2006 and references there).

Although valid concerns, they do not rule out a fundamentally deterministic learning mechanisms which possesses nondeterministic components (e.g., a system that requires multiple exposures to an unambiguous E-trigger, or even exposure to multiple E-triggers multiple times, in order to ensure reliable learning in the face of noise). Such a learning system could well model the course of acquisition better than a fully trial-and-error system (See discussion above concerning the fit of Yang's model to developmental data).

Another concern that *is* specific to unambiguous E-triggers is their implementation – are they an innately endowed list of fully formed sentence patterns? Unlikely. A list of innately endowed "schemas" (e.g., a prepositional object separated from its preposition)? Possibly. Or are they computationally derived from innately endowed I-triggers? This last is the most attractive of the three alternatives. However, it remains an open question if it is a computationally feasible option (Gibson & Wexler, 1994; Sakas & Fodor, Submitted).

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

Still the advantages of a deterministic, triggering learning system are compelling. One would be hard pressed to imagine a more efficient and precise acquisition model than one that uses unambiguous (including disambiguated) E-triggers to set a parameter to its correct value. Equally as important, deterministic learning allows for reliable use of default values. Default values are necessary if the marked value of a parameter generates a superset of the default value; they are a straightforward implementation of the Subset Principle in a parametric system. They also minimize the workload of the learning mechanism since no computations need to be carried out to establish a default value. As a result, defaults are often presupposed in much of pyscholinguistics research. However, if the learner employs a nondeterministic search strategy, defaults are rendered virtually worthless. This is because a trial-and-error learner is never completely certain that the current hypothesized grammar is correct; i.e., the hypothesis might contain incorrect marked values for any number of parameters; the search heuristics would need sufficient evidence for both values of every parameter which in turns entails less efficient learning than a learning model that makes use of defaults. Although there have been attempts aimed at trying to reap the benefits of default values within a nondeterministic learning paradigm (Bertolo, 1995b, a), the intricate solutions are unsatisfactory compared to the simplicity of the classic triggering model; a default value can be reliably left unaltered unless unambiguous evidence for the marked value is encountered by the learner.

In the end, it appears (at least to me) that the benefits of a deterministic, classical triggering model outweigh the disadvantages and that many of the components of classical triggering should be, and could be, retained in computational modeling. But this endeavor is in its infancy. *If* I were to chart an 'ideal' path that the next generation of computational models of syntactic parameter setting would take, I'd mark first that modelers need to develop a psychologically plausible computational mapping from I-triggers to E-triggers. This is not a trivial endeavor and has remained an open question for several decades, however, any computational model of triggering will need to implement mapping between I- and E- triggers if it is to be widely accepted as a model of child language acquisition. Following this, extensive study of parameter interaction and parametric ambiguity would need to be conducted in order to discover if trigger disambiguation can overcome whatever pockets in the domain of human languages exhibit a dearth of unambiguous E-triggers. To the extent that trigger disambiguation fails in creating sufficient unambiguous triggers, the learning model would employ a simple statistical mechanism (e.g., Fodor, 1998b; Yang, 2002) to navigate a (expectantly) small area of the parameter space non-deterministically (see discussion of the Guessing-STL models above). Finally, robustness would need to be built in the model and tested against corpora of actual child-directed speech complete with all expected mis-speaks, sentence fragments, etc. – and, of course, in as many languages as possible. That is, *if* I were to map such a trajectory out!

This future is clearly 'a ways away', but going back to the essential elements of classical triggering theory, adapting them in a computational implementation of parameter setting, and, finally empirically demonstrating that the adaptations make for an efficient, precise, and developmentally accurate model, is the quickest and most promising route to get there.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

4 References

Battistella, E. (1996). *The Logic of Markedness*: Oxford University Press, USA.

Bertolo, S. (1995a). *Learnability Properties of Parametric Models for Natural Language Acquisition.* Doctoral thesis, Rutgers University.

Bertolo, S. (1995b). Maturation and learnability in parametric systems. *Language Acquisition, 4*(4), 277-318.

Bertolo, S., Broihier, K., Gibson, E., & Wexler, K. (1997a). Characterizing learnability conditions for cue-based learners in parametric language systems. In *Proceedings of The Fifth meeting of The Mathematics of Language Conference (MOL5) (www.dfki.de/events/mol)* Saarbrücken.

Bertolo, S., Broihier, K., Gibson, E., & Wexler, K. (1997b). Cue-based learners in parametric language systems: Application of general results to a recently proposed learning algorithm based on unambiguous 'superparsing'. In M. G. Shafto & P. Langley (Eds.), *Proceedings of The Nineteenth Annual Conference of the Cognitive Science Society (CogSci-1997)*, Stanford University.

Bertolo, S. (2001). A brief overview of learnability. In S. Bertolo (Ed.), *Language Acquisition and Learnability* (pp. 1-14). Cambridge: Cambridge University Press.

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

Berwick, R. C., & Niyogi, P. (1996). Learning from triggers. *Linguistic Inquiry, 27*(4), 605-622.

Briscoe, E. J. (2000). Grammatical acquisition: Inductive bias and coevolution of language and the language acquisition device. *Language, 76*(2), 245-296.

Brown, R., & Hanlon, C. (1970). Derivational complexity and order of acquisition in child speech. In J. Hayes (Ed.), *Cognition and the Development of Language* (pp. 11-53). New York: Wiley.

Bush, R. R., & Mosteller, F. (1955). *Stochastic models for learning*. New York,: Wiley.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. Cambridge: MIT Press.

Chomsky, N. (1981). *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, N. (1986). *Knowledge of Language: Its Nature, Origin, and Use*. New York: Praeger.

Chomsky, N. (1988). *Language and problems of knowledge: The Managua lectures*: The MIT press.

Chomsky, N. (2000). *New horizons in the study of language and mind*. Cambridge [England] ; New York: Cambridge University Press.

Clark, R. (1989). On the relationship between the input data and parameter setting. In J. Carter & R. M. Dechaine (Eds.), *Proceedings of The 19th Annual Meeting of the North East Linguistic Society (NELS 19)*, Amherst, MA.

Clark, R. (1992). The selection of syntactic knowledge. *Language Acquisition, 2*(2), 83-149.

Clark, R., & Roberts, I. (1993). A computational model of language learnability and language change. *Linguistic Inquiry, 24*(2), 299-345.

Daelemans, W. (2002). Review of: Knowledge and Learning in Language, Charles D. Yang. *Glot International, 6*(5), 137-142.

Deprez, V., & Pierce, A. (1993). Negation and functional projections in early grammar. *Linguistic Inquiry, 24*(1), 25-67.

Dresher, B. E., & Kaye, J. D. (1990). A computational learning model for metrical phonology. *Cognition, 34*(2), 137-195.

Dresher, B. E. (1999). Charting the learning path: Cues to parameter setting. *Linguistic Inquiry, 30*(1), 27-67.

Fodor, J. D. (1998a). Unambiguous triggers. *Linguistic Inquiry, 29*(1), 1-36.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

Fodor, J. D. (1998b). Parsing to learn. *Journal of Psycholinguistic Research, 27*(3), 339-374.

Fodor, J. D., & Sakas, W. G. (2004). Evaluating models of parameter setting. In A. Brugos, L. Micciulla & C. E. Smith (Eds.), *Proceedings of The 28th Annual Boston University Conference on Language Development (BUCLD 28)*, Boston, MA.

Fodor, J. D., & Sakas, W. G. (2005). The Subset Principle in syntax: Costs of compliance. *Journal of Linguistics, 41*(03), 513-569.

Frank, R., & Kapur, S. (1996). On the use of triggers in parameter setting. *Linguistic Inquiry, 27*(4), 623-660.

Gibson, E., & Wexler, K. (1994). Triggers. *Linguistic Inquiry, 25*(3), 407-454.

Gibson, E. A. F. (1991). *A computational theory of human linguistic processing: Memory limitations and processing breakdown.* Doctoral thesis, Carnegie Mellon University.

Gold, E. M. (1967). Language identification in the limit. *Information and Control, 10*(5), 447-474.

Kapur, S. (1994). Some applications of formal learning theory results to natural language acquisition. In B. Lust & G. Hermon (Eds.), *Syntactic Theory and First Language Acquisition: Cross-linguistic Perspectives. Volume 2: Binding, Dependencies, and Learnability* (pp. 491-508). Hillsdale, NJ: Lawrence Erlbaum.

Kohl, K. T. (1999). *An Analysis of Finite Parameter Learning in Linguistic Spaces.* Masters thesis, Massachusetts Institute of Technology.

Lightfoot, D. (1999). *The Development of Language: Acquisition, Change, and Evolution*. Malden, MA: Blackwell.

Lightfoot, D. (2006). *How New Languages Emerge*. Cambridge: Cambridge University Press.

MacWhinney, B. (2000). *The CHILDES Project: Tools for Analyzing Talk. Third Edition*. Mahwah, NJ: Lawrence Erlbaum Associates.

Manzini, M. R., & Wexler, K. (1987). Parameters, binding theory, and learnability. *Linguistic Inquiry, 18*(3), 413-444.

Marcus, G. (1993). Negative evidence in language acquisition. *Cognition, 46*(1), 53-85.

Niyogi, P., & Berwick, R. (1996). A language learning model for finite parameter spaces. *Cognition, 61*(1-2), 161-193.

Nyberg, E. H., III. (1992). *A Non-Deterministic, Success-Driven Model of Parameter Setting in Language Acquisition.* Doctoral thesis, Carnegie Mellon University.

Osherson, D. N., Stob, M., & Weinstein, S. (1986). *Systems that learn: An Introduction to Learning Theory for Cognitive and Computer scientists*. Cambridge, MA: MIT Press.

Pearl, L. S. (2007). *Necessary Bias in Natural Language Learning.* Doctoral thesis, University of Maryland.

Pinker, S. (1979). Formal models of language learning. *Cognition, 7*(3), 217-283.

Sakas, W. G. (2000). *Ambiguity and the Computational Feasibility of Syntax Acquisition.* Doctoral thesis, City University of New York, New York, NY.

Sakas, W. G., & Fodor, J. D. (2001). The Structural Triggers Learner. In S. Bertolo (Ed.), *Language Acquisition and Learnability* (pp. 172-233). Cambridge: Cambridge University Press.

Sakas, W. G. (2003). A Word-Order Database for Testing Computational Models of Language Acquisition. In E. Hinrichs & D. Roth (Eds.), *Proceedings of The 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, Sapporo, Japan.

Sakas, W. G., & Fodor, J. D. (Submitted). Disambiguating Syntactic Triggers.

Snyder, W. (2007). *Child Language: The Parametric Approach*. Oxford: Oxford University Press.

Straus, K. (2008). *Validations of a Probabilistic Model of Language Learning.* Doctoral thesis, Northeastern, Boston, MA.

To appear, Lidz, J., W. Snyder & J. Pater, eds. Oxford Handbook of Developmental Linguistics. Oxford University Press

Sugisaki, K., & Snyder, W. (2003). Do parameters have default values?: Evidence from the acquisition of English and Spanish. In Y. Otsu (Ed.), *Proceedings of The Fourth Tokyo Conference on Psycholinguistics*, Hituzi Syobo, Tokyo.

Sugisaki, K., & Snyder, W. (2005). Evaluating the variational model of language acquisition. In K. U. Deen, J. Nomura, B. Schulz & B. D. Schwartz (Eds.), *Proceedings of The Inaugural Conference on Generative Approaches to Language Acquisition - North America (GALANA)*, University of Hawaii.

Turkel, W. J. (1996). Acquisition by a genetic algorithm-based model in spaces with local maxima. *Linguistic Inquiry, 27*(2), 350-355.

van Kampen, J. (1997). *First Steps in Wh-movement*. Delft: Eburon.

Villavicencio, A. (2001). *The Acquisition of a Unification-Based Generalised Categorial Grammar.* Doctoral thesis, Cambridge University.

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

Wexler, K., & Manzini, R. (1987). Parameters and learnability in binding theory. In T. Roeper & E. Williams (Eds.), *Parameter Setting* (pp. 41-76). Dordrecht: Reidel.

Yang, C. D. (2002). *Knowledge and Learning in Natural Language*. Oxford: Oxford University Press.