A Word-Order Database for Testing
Computational Models of Language Acquisition

William Gregory Sakas
Department of Computer Science
PhD Programs in Linguistics and Computer Science
Hunter College and The Graduate Center
City University of New York
sakas@hunter.cuny.edu

Abstract

An investment of effort over the last two years has begun to produce a wealth of data concerning computational psycholinguistic models of syntax acquisition. The data is generated by running simulations on a recently completed database of word order patterns from over 3,000 abstract languages. This article presents the design of the database which contains sentence patterns, grammars and derivations that can be used to test acquisition models from widely divergent paradigms. The domain is generated from grammars that are linguistically motivated by current syntactic theory and the sentence patterns have been validated as psychologically/developmentally plausible by checking their frequency of occurrence in corpora of child-directed speech. A small case-study simulation is also presented.

1   Introduction

The exact process by which a child acquires the grammar of his or her native language is one of the most beguiling open problems of cognitive science. There has been recent interest in computer simulation of the acquisition process and the interrelationship between such models and linguistic and psycholinguistic theory. The hope is that through computational study, certain bounds can be established which may be brought to bear on pivotal issues in developmental psycholinguistics.

Simulation research is a significant departure from standard learnability models that provide results through formal proof (e.g., Bertolo, 2001;

Gold, 1967; Jain et al., 1999; Niyogi, 1998; Niyogi & Berwick, 1996; Pinker, 1979; Wexler & Culicover, 1980, among many others). Although research in learnability theory is valuable and ongoing, there are several disadvantages to formal modeling of language acquisition:

- Certain proofs may involve impractically many steps for large language domains (e.g. those involving Markov methods).
- Certain paradigms are too complex to readily lend themselves to deductive study (e.g. connectionist models).[1]
- Simulations provide data on intermediate stages whereas formal proofs typically prove whether a domain is (or more often is *not*) learnable a priori to specific trials.
- Proofs generally require simplifying assumptions which are often distant from natural language.

However, simulation studies are not without disadvantages and limitations. Most notable perhaps, is that out of practicality, simulations are typically carried out on small, severely circumscribed domains – usually just large enough to allow the researcher to hone in on how a particular model (e.g. a connectionist network or a principles & parameters learner) handles a few grammatical features (e.g. long-distance agreement and/or topicalization) often, though not always, in a single language. So although there have been many successful studies that demonstrate how one algorithm or another is able to acquire some aspect of grammatical structure, there is little doubt that the question of what mechanism children actually employ during the acquisition process is still open.

This paper reports the development of a large, multilingual database of sentence patterns, gram-

---

[1] Although see Niyogi, 1998 for some insight.

mars and derivations that may be used to test computational models of syntax acquisition from widely divergent paradigms. The domain is generated from grammars that are linguistically motivated by current syntactic theory and the sentence patterns have been validated as psychologically/developmentally plausible by checking their frequency of occurrence in corpora of child-directed speech. We report here the structure of the domain, its interface and a case-study that demonstrates how the domain has been used to test the feasibility of several different acquisition strategies.

The domain is currently publicly available on the web via http://146.95.2.133 and it is our hope that it will prove to be a valuable resource for investigators interested in computational models of natural language acquisition.

2 The Language Domain Database

The focus of the *language domain database*, (hereafter *LDD*), is to make readily available the different word order patterns that children are typically exposed to, together with all possible syntactic derivations of each pattern. The patterns and their derivations are generated from a large battery of grammars that incorporate many features from the domain of natural language.

At this point the multilingual language domain contains sentence patterns and their derivations generated from 3,072 abstract grammars. The patterns encode sentences in terms of tokens denoting the grammatical roles of words and complex phrases, e.g., subject (S), direct object (O1), indirect object (O2), main verb (V), auxiliary verb (Aux), adverb (Adv), preposition (P), etc. An example pattern is *S Aux V O1* which corresponds to the English sentence: *The little girl can make a paper airplane*. There are also tokens for topic and question markers for use when a grammar specifies overt topicalization or question marking.

Declarative sentences, imperative sentences, negations and questions are represented within the LDD, as is prepositional movement/stranding (*pied-piping*), null subjects, null topics, topicalization and several types of movement.

Although more work needs to be done, a first round study of actual child-directed sentences from the CHILDES corpus (MacWhinney, 1995) indicates that our patterns capture many sentential word orders that children typically encounter in the period from 1-1/2 to 2-1/2 years; the period generally accepted by psycholinguists to be when children establish the correct word order of their native language. For example, although the LDD is currently limited to degree-0 (i.e. no embedding) and does not contain DP-internal structure, after examining by hand, several thousand sentences from corpora in the CHILDES database in five languages (English, German, Italian, Japanese and Russian), we found that approximately 85% are degree-0 and an approximate 10 out of 11 have no internal DP structure.

Adopting the principles and parameters (*P&P*) hypothesis (Chomsky, 1981) as the underlying framework, we implemented an application that generated patterns and derivations given the following points of variation between languages:

1. Affix Hopping	2. Comp Initial/Final
3. I to C Movement	4. Null Subject
5. Null Topic	6. Obligatory Topic
7. Object Final/Initial	8. Pied Piping
9. Question Inversion	10. Subject Initial/Final
11. Topic Marking	12. V to I Movement
13. Obligatory Wh movement	

The patterns have fully specified X-bar structure, and movement is implemented as HPSG local dependencies. Pattern production is generated top-down via rules applied at each subtree level. Subtree levels include: CP, C', IP, I', NegP, Neg', VP, V' and PP. After the rules are applied, the subtrees are fully specified in terms of node categories, syntactic feature values and constituent order. The subtrees are then combined by a simple unification process and syntactic features are percolated down. In particular, movement chains are represented as traditional "slash" features which are passed (locally) from parent to daughter; when unification is complete, there is a trace at the bottom of each slash-feature path. Other features include +/-NULL for non-audible tokens (e.g. S[+NULL] represents a null subject *pro*), +TOPIC to represent a topicalized token, +WH to represent "who", "what", etc. (or "qui", "que" if one prefers), +/-FIN to mark if a verb is tensed or not and the illocutionary (ILLOC) features Q, DEC, IMP for questions, declaratives and imperatives respectively.

Although further detail is beyond the scope of this paper, those interested may refer to Fodor et al. (2003) which resides on the LDD website.

It is important to note that the domain is suitable for many paradigms beyond the P&P framework. For example the context-free rules (with local dependencies) could be easily extracted and used to test probabilistic CFG learning in a multilingual domain. Likewise the patterns, without their derivations, could be used as input to statistical/connectionist models which eschew traditional (generative) structure altogether and search for regularity in the left-to-right strings of tokens that makeup the learner's input stream. Or, the patterns could help bootstrap the creation of a domain that might be used to test particular types of lexical learning by using the patterns as templates where tokens may be instantiated with actual words from a lexicon of interest to the investigator. The point is that although a particular grammar formalism was used to generate the patterns, the patterns are valid independently of the formalism that was in play during generation.[2]

To be sure, similar domains have been constructed. The relationship between the LDD and other artificial domains is summarized in Table 1.

In designing the LDD, we chose to include syntactic phenomena which:

i) occur in a relatively high proportion of the known natural languages;

ii) are frequently exemplified in speech directed to 2-year-olds;

iii) pose potential learning problems (e.g. cross-language ambiguity) for which theoretical solutions are needed;

iv) have been a focus of linguistic and/or psycholinguistic research;

v) have a syntactic analysis that is broadly agreed on.

As a result the following have been included:

- By criteria (i) and (ii): negation, non-declarative sentences (questions, imperatives).
- By criterion (iv): null subject parameter (Hyams 1986 and since).
- By criterion (iv): affix-hopping (though not widespread in natural languages).
- By criterion (v): no scrambling yet.

There are several phenomena that the LDD does not yet include:

- No verb subcategorization.
- No interface with LF (cf. Briscoe 2000; Villavicencio 2000).
- No discourse contexts to license sentence fragments (e.g., DP or PP fragments).
- No XP-internal structure yet (except PP = P + O3, with piping or stranding).
- No Linear Correspondence Axiom (Kayne 1994).
- No feature checking as implementation of movement parameters (Chomsky 1995).

	# parameters	# languages	Tree structure?	Language properties
Gibson & Wexler (1994)	3	8	Not fully specified	Word order, V2
Bertolo et. al (1997b)	7	64 distinct	Yes	G&W + V-raising to Agr, T; deg-2
Kohl (1999) based on Bertolo	12	2,304	Partial	Bertolo et al. (1997b) + scrambling
Sakas & Nishimoto (2002)	4	16	Yes	G&W + null subject/topic
LDD	**13**	**3,072**	**Yes**	**S&N + wh-movt + imperatives +aux inversion, etc.**

Table 1: A history of abstract domains for word-order acquisition modeling.

---

[2] If this is the case, one might ask: Why bother with a grammar formalism at all; why not use actual child-directed speech as input instead of artificially generated patterns? Although this approach has proved workable for several types of non-generative acquisition models, a generative (or hybrid) learner is faced with the task of selecting the rules or parameter values that generate the linguistic environment being encountered by the learner. In order to simulate this, there must be some grammatical structure incorporated into the experimental design that serves as the target the learner must acquire. Constructing a viable grammar and a parser with coverage over a multilingual domain of real child-directed speech is a daunting proposition. Even building a parser to parse a *single* language of child-directed speech turns out to be extremely difficult. See, for example, Sagae, Lavie, & MacWhinney (2001), which discusses an impressive number of practical difficulties encountered while attempting to build a parser that could cope with the EVE corpus; one the cleanest transcriptions in the CHILDES database. By abstracting away from actual child-directed speech, we were able to build a pattern generator and include the pattern derivations in the database for retrieval during simulation runs, effectively sidestepping the need to build an online multilingual parser.

The LDD on the web: The two primary purposes of the web-interface are to allow the user to interactively peruse the patterns and the derivations that the LDD contains and to download raw data for the user to work with locally.

Users are asked to register before using the LDD online. The user ID is typically an email address, although no validity checking is carried out. The benefit of entering a valid email address is simply to have the ability to recover a forgotten password, otherwise a user can have full access anonymously.

The interface has three primary areas: *Grammar Selection*, *Sentence Selection* and *Data Download*. First a user has to specify, on the Grammar Selection page, which settings of the 13 parameters are of interest and save those settings as an available grammar. A user may specify multiple grammars. Then in the sentence selection page a user may peruse sentences and their derivations. On this page a user may annotate the patterns and derivations however he or she wishes. All grammar settings and annotations are saved and available the next time the user logs on. Finally on the Data Download page, users may download data so that they can use the patterns and derivations offline.

The derivations are stored as bracketed strings representing tree structure. These are practically indecipherable by human users. E.g.:

(CP[ILLOC Q][+FIN][+WH] "Adv[+TOPIC]" (Cbar[ILLOC Q] [+FIN][+WH][SLASH Adv](C[ILLOC Q][+FIN] "KA") (IP[ILLOC Q][+FIN][+WH][SLASH Adv]"S" (Ibar[ILLOC Q][+FIN][+WH][SLASH Adv](I[ILLOC Q][+FIN]"Aux[+FIN]")(NegP[+WH] [SLASH Adv](NegBar[+WH][SLASH Adv](Neg "NOT") (VP[+WH][SLASH Adv](Vbar[+WH][SLASH Adv](V"Verb")"O1" "O2" (PP[+WH] "P" "O3[+WH]")"Adv[+NULL][SLASH Adv]"))))))))

To be readable, the derivations are displayed graphically as tree structures. Towards this end we have utilized a set of publicly available LaTex macros: *QTree* (Siskind & Dimitriadis, [online]). A server-side script parses the bracketed structures into the proper QTree/LaTex format from which a pdf file is generated and subsequently sent to the user's client application.

Even with the graphical display, a simple sentence-by-sentence presentation is untenable given the large amount of linguistic data contained in the database. The Sentence Selection area allows users to access the data filtered by sentence type and/or by grammar features (e.g. all sentences that have obligatory-wh movement and contain a prepositional phrase), as well as by the user's defined grammar(s) (all sentences that are "Italian-like").

On the Data Download page, users may filter sentences as on the Sentence Selection page and download sentences in a tab-delimited format. The entire LDD may also be downloaded – approximately 17 MB compressed, 600 MB as a raw ascii file.

3 A Case Study: Evaluating the efficiency of parameter-setting acquisition models.

We have recently run experiments of seven parameter-setting (P&P) models of acquisition on the domain. What follows is a brief discussion of the algorithms and the results of the experiments. We note in particular where results stemming from work with the LDD lead to conclusions that differ from those previously reported. We stress that this is not intended as a comprehensive study of parameter-setting algorithms or acquisition algorithms in general. There is a large number of models that are omitted; some of which are targets of current investigation. Rather, we present the study as an example of how the LDD could be effectively utilized.

In the discussion that follows we will use the terms "pattern", "sentence" and "input" interchangeably to mean a left-to-right string of tokens drawn from the LDD without its derivation.

3.1 A Measure of Feasibility

As a simple example of a learning strategy and of our simulation approach, consider a domain of 4 binary parameters and a memoryless learner [3] which blindly guesses how all 4 parameters should be set upon encountering an input sentence. Since there are 4 parameters, there are 16 possible combinations of parameter settings. i.e., 16 different grammars. Assuming that each of the 16 grammars is equally likely to be guessed, the learner will consume, on average, 16 sentences before achieving the target grammar. This is one measure of a model's efficiency or *feasibility*.

---

[3] By "memoryless" we mean that the learner processes inputs one at a time without keeping a history of encountered inputs or past learning events.

However, when modeling natural language acquisition, since practically all human learners attain the target grammar, the average number of expected inputs is a less informative statistic than the expected number of inputs required for, say, 99% of all simulation trials to succeed. For our blind-guess learner, this number is 72.[4] We will use this 99-percentile feasibility measure for most discussion that follows, but also include the average number of inputs for completeness.

3.2 The Simulations

In all experiments:
- The learners are memoryless.
- The language *input sample* presented to the learner consists of only grammatical sentences generated by the target grammar.
- For each learner, 1000 trials were run for each of the 3,072 target languages in the LDD.
- At any point during the acquisition process, each sentence of the target grammar is equally likely to be presented to the learner.

Subset Avoidance and Other Local Maxima: Depending on the algorithm, it may be the case that a learner will never be motivated to change its current hypothesis (G_{curr}), and hence be unable to ultimately achieve the target grammar (G_{targ}). For example, most error-driven learners will be trapped if G_{curr} generates a language that is a superset of the language generated by G_{targ}. There is a wealth of learnability literature that addresses local maxima and their ramifications.[5] However, since our study's focus is on feasibility (rather than on whether a domain is learnable given a particular algorithm), we posit a built-in avoidance mechanism, such as the *subset principle* and/or *default values* that preclude local maxima; hence, we set aside trials where a local maximum ensues.

3.3 The Learners' strategies

In all cases the learner is *error-driven*: if G_{curr} can parse the current input pattern, retain it.[6]

The following refers to what the learner does when G_{curr} fails on the current input.

- **Error-driven, blind-guess (EDBG):** adopt any grammar from the domain chosen at random – not psychologically plausible, it serves as our baseline.
- **TLA** (Gibson & Wexler, 1994)**:** change any *one* parameter value of those that make up G_{curr}. Call this new grammar G_{new}. If G_{new} can parse the current input, adopt it. Otherwise, retain G_{curr}.
- **Non-Greedy TLA** (Niyogi & Berwick, 1996)**:** change any *one* parameter value of those that make up G_{curr}. Adopt it. (I.e. there is no testing of the new grammar against the current input).
- **Non-SVC TLA** (Niyogi & Berwick, 1996)**:** try *any* grammar in the domain. Adopt it only in the event that it can parse the current input.
- **Guessing STL** (Fodor, 1998a)**:** Perform a structural parse of the current input. If a choice point is encountered, chose an alternative based on one of the following and then set parameter values based on the final parse tree:
  - STL *Random Choice* (*RC*) – randomly pick a parsing alternative.
  - *Minimal Chain* (*MC*) – pick the choice that obeys the *Minimal Chain Principle* (De Vincenzi, 1991), i.e., avoid positing movement transformations if possible.
  - *Local Attachment/Late Closure* (*LAC*) –pick the choice that attaches the new word to the current constituent (Frazier, 1978).

The EDBG learner is our first learner of interest. It is easy to show that the average and 99% scores increase exponentially in the number of parameters and syntactic research has proposed more than 100 (e.g. Cinque, 1999). Clearly, human learners do not employ a strategy that performs as poorly as this. Results will serve as a baseline to compare against other models.

---

[4] The average and 99-percentile figures (16 and 72) in this section are easily derived from the fact that input consumption follows a hypergeometric distribution.

[5] Discussion of the problem of subset relationships among languages starts with Gold's (1967) seminal paper and is discussed in Berwick (1985) and Wexler & Manzini (1987). Detailed accounts of the types of local maxima that the learner might encounter in a domain similar to the one we employ are given in Frank & Kapur (1996), Gibson & Wexler (1994), and Niyogi & Berwick (1996).

[6] We intend for a "*can-parse/can't-parse* outcome" to be equivalent to the result from a language membership test. If the current input sentence is one of the set of sentences generated by G_{curr}, *can-parse* is engendered; if not, *can't-parse*.

	99%	Average
EDBG	16,663	3,589

Table 2: EDBG, # of sentences consumed

The TLA: The TLA incorporates two search heuristics: the *Single Value Constraint* (*SVC*) and *Greediness*. In the event that G_{curr} cannot parse the current input sentence *s*, the TLA attempts a second parse with a randomly chosen new grammar, G_{new}, that differs from G_{curr} by exactly one parameter value (SVC). If G_{new} can parse *s*, G_{new} becomes the new G_{curr} otherwise G_{new} is rejected as a hypothesis (Greediness). Following Berwick and Niyogi (1996), we also ran simulations on two variants of the TLA – one with the Greediness heuristic but without the SVC (TLA minus SVC, *TLA–SVC*) and one with the SVC but without Greediness (TLA minus Greediness, *TLA–Greed*). The TLA has become a seminal model and has been extensively studied (cf. Bertolo, 2001 and references therein; Berwick & Niyogi, 1996; Frank & Kapur, 1996; Sakas, 2000; among others). The results from the TLA variants operating in the LDD are presented in Table 3.

	99%	Average
TLA-SVC	67,896	11,273
TLA-Greed	19,181	4,110
TLA	16,990	961

Table 3: TLA variants, # of sentences consumed

Particularly interesting is that contrary to results reported by Niyogi & Berwick (1996) and Sakas & Nishimoto (2002), the SVC and Greediness constraints *do* help the learner achieve the target in the LDD. The previous research was based on simulations run on much smaller 9 and 16 language domains (see Table 1). It would seem that the local hill-climbing search strategies employed by the TLA do improve learning efficiency in the LDD. However, even at best, the TLA performs less well than the blind guess learner. We conjecture that this fact probably rules out the TLA as a viable model of human language acquisition.

The STL: Fodor's *Structural Triggers Learner* (*STL*) makes greater use of the parser than the TLA. A key feature of the model is that parameter values are not simply the standardly presumed 0 or 1, but rather bits of tree structure or *treelets*. Thus, a grammar, in the STL sense, is a collection of treelets rather than a collection of 1's and 0's. The STL is error-driven. If G_{curr} cannot license *s*, new treelets will be utilized to achieve a successful parse.[7] 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, retain the treelets that make up G_{curr}.
2. Otherwise, parse the sentence making use of any or all parametric treelets available 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. The TLA variants do not have this capacity because they rely only on a can-parse/can't-parse outcome and do 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* (*SSTL*) and the *weak STL*.

The SSTL 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 SSTL an extremely powerful, albeit psychologically implausible, learner.[8] However, this is not necessarily the case. The SSTL 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 yields an unambiguous treelet for a particular parameter.

---

[7] In addition to the treelets, UG principles are also available for parsing, as they are in the other models discussed above.
[8] It is important to note that Fodor (1998a) does not put forth the strong STL as a psychologically plausible model. Rather, it is intended to demonstrate the potential effectiveness of parametric decoding.

Unlike the SSTL, the weak STL executes a psychologically plausible left-to-right serial (deterministic) parse. One variant of the weak STL, the *waiting STL* (*WSTL*), 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.

As with the TLA, these STL variants have been studied from a mathematical perspective (Bertolo et al., 1997a; Sakas, 2000). Mathematical analyses point to the fact that the strong and weak STL are extremely efficient learners in conducive domains with some unambiguous inputs but may become paralyzed in domains with high degrees of ambiguity. These mathematical analyses among other considerations spurred a new class of weak STL variants which we informally call the *guessing STL* family.

The basic idea behind the guessing STL models is that there is some information available even in sentences that are ambiguous, and some strategy that can exploit that information. We incorporate three different heuristics into the original STL paradigm, the RC, MC and LAC heuristics described above.

Although the MC and LAC heuristics are not stochastic, we regard them as "guessing" heuristics because, unlike the WSTL, a learner cannot be certain that the parametric treelets obtained from a parse guided by MC and LAC are correct for the target. These heuristics are based on well-established human parsing strategies. Interestingly, the difference in performance between the three variants is slight. Although we have just begun to look at this data in detail, one reason may be that the typical types of problems these parsing strategies address are not included in the LDD (e.g. relative clause attachment ambiguity). Still, the STL variants perform the most efficiently of the strategies presented in this small study (approximately a 100-fold improvement over the TLA). Certainly this is due to the STL's ability to perform parametric decoding. See Fodor (1998b) and Sakas & Fodor (2001) for detailed discussion about the power of decoding when applied to the acquisition process.

Guessing STL	99%	Average
RC	1,486	166
MC	1,412	160
LAC	1,923	197

Table 4: guessing STL family, # of sentences consumed

4 Conclusion and future work

The thrust of our current research is directed at collecting data for a comprehensive, comparative study of psycho-computational models of syntax acquisition. To support this endeavor, we have developed the Language Domain Database – a publicly available test-bed for studying acquisition models from diverse paradigms.

Mathematical analysis has shown that learners are extremely sensitive to various distributions in the input stream (Niyogi & Berwick, 1996; Sakas, 2000, 2003). Approaches that thrive in one domain may dramatically flounder in others. So, whether a particular computational model is successful as a model of natural language acquisition is ultimately an empirical issue and depends on the exact conditions under which the model performs well and the extent to which those favorable conditions are in line with the facts of human language. The LDD is a useful tool that can be used within such an empirical research program.

Future work: Though the LDD has been validated against CHILDES data in certain respects, we intend to extend this work by adding distributions to the LDD that correspond to actual distributions of child-directed speech. For example, what percentage of utterances, in child-directed Japanese, contain pro-drop? object-drop? How often in English does the pattern: *S[+WH] aux Verb O1* occur and at what periods of a child's development? We believe that these distributions will shed some light on many of the complex subtleties involved in ambiguity disambiguation and the role of nondeterminism and statistics in the language acquisition process. This is proving to be a formidable, yet surmountable task; one that we are just beginning to tackle.

References

Bertolo, S. (Ed.) (2001). *Language Acquisition and Learnability*. Cambridge, UK: Cambridge University Press.

Bertolo, S., Broihier, K., Gibson, E., & Wexler, K. (1997a). Characterizing learnability conditions for cue-based learners in parametric language systems. *Proceedings of the Fifth Meeting on Mathematics of Language*.

Bertolo, S., Broihier, K., Gibson, E., and 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 and P. Langley (eds.) *the Cognitive Science Society*, Mahwah NJ: Lawrence Erlbaum Associates.

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

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

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

Chomsky, N. (1995) *The Minimalist Program*. Cambridge MA: MIT Press.

Cinque, G. (1999) *Adverbs and Functional Heads.* Oxford Oxford, UK:University Press, Oxford, UK.

Fodor, J. D. (1998a) Unambiguous triggers, *Linguistic Inquiry* 29.1, 1-36.

Fodor, J. D. (1998b) Parsing to learn. *Journal of Psycholinguistic Research* 27.3, 339-374.

Fodor, J.D., Melnikova, Y. & Troseth, E. (2002) A structurally defined language domain for testing syntax acquisition models. Technical Report. CUNY Graduate Center.

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

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

Hyams, N. (1986) *Language Acquisition and the Theory of Parameters*. Dordrecht: Reidel.

Jain, S., E. Martin, D. Osherson, J. Royer, and A. Sharma. (1991) *Systems That Learn*. 2nd ed. Cambridge, MA: MIT Press.

Kayne, R. S. (1994) *The Antisymmetry of Syntax.* Cambridge MA: MIT Press.

Kohl, K.T. (1999) *An Analysis of Finite Parameter Learning in Linguistic Spaces*. Master's Thesis, MIT.

MacWhinney, B. (1995) *The CHILDES Project: Tools for Analyzing Talk*. (2nd ed.) Hillsdale, NJ: Lawrence Erlbaum Associates.

Niyogi, P (1998) *The Informational Complexity of Learning: Perspectives on Neural Networks and Generative Grammar* Dordrecht: Kluwer Academic.

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

Sagae, K., Lavie, A., MacWhinney, B. (2001) Parsing the CHILDES database: Methodology and lessons learned. In Proceedings of the Seventh International Workshop in Parsing Technologies. Beijing, China.

Sakas, W.G. (in prep) Grammar/Language smoothness and the need (or not) of syntactic parameters. Hunter College and The Graduate Center, City University of New York.

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

Sakas, W.G. and Fodor, J.D. (2001). The Structural Triggers Learner. In S. Bertolo (ed.) *Language Acquisition and Learnability*. Cambridge, UK: Cambridge University Press.

Sakas, W.G. and Nishimoto, E. (2002) Search, Structure or Statistics? A Comparative Study of Memoryless Heuristics for Syntax Acquisition, *Proceedings of the 24th Annual Conference of the Cognitive Science Society*. Hillsdale NJ: Lawrence Erlbaum Associates,

Siskind, J.M & Dimitriadis, A., [Online 5/20/2003] *Documentation for qtree, a LaTex tree package* http://www.ling.upenn.edu/advice/latex/qtree/

Villavicencio, A. (2000) The use of default unification in a system of lexical types. Paper presented at the Workshop on Linguistic Theory and Grammar Implementation, Birmingham,UK.

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