

Automatic Inference of Minimalist Grammars using an SMT-Solver

Sagar Indurkha

MIT

INDURKS@MIT.EDU

Abstract

We introduce (1) a novel parser for Minimalist Grammars (MG), encoded as a system of first-order logic formulae that may be evaluated using an SMT-solver, and (2) a novel procedure for inferring Minimalist Grammars using this parser. The input to this procedure is a sequence of sentences that have been annotated with syntactic relations such as semantic role labels (connecting arguments to predicates) and subject-verb agreement. The output of this procedure is a set of minimalist grammars, each of which is able to parse the sentences in the input sequence such that the parse for a sentence has the same syntactic relations as those specified in the annotation for that sentence. We applied this procedure to a set of sentences annotated with syntactic relations and evaluated the inferred grammars using cost functions inspired by the Minimum Description Length principle and the Subset principle. Inferred grammars that were *optimal* with respect to certain combinations of these cost functions were found to align with contemporary theories of syntax.

Keywords: Minimalist Grammar, Satisfiability Modulo Theory, Grammatical Inference

1. Introduction

Inspired by earlier formulations of grammars using logic (Pereira and Warren (1983); Rayner et al. (1988); Stabler (1993); Rogers (1998); Graf (2013)) and recent, substantive improvements in the performance of SMT-solvers (De Moura and Bjørner (2011); Cadar and Sen (2013)), we have developed a novel procedure, with the form of a model of language acquisition (Chomsky (1965)), for automatically inferring Minimalist Grammars (MG) (Stabler (1996)) from a sequence of sentences that have been annotated with syntactic relations for predicate-argument structure and morphological agreement.¹ In this study, we report preliminary results that demonstrate our inference procedures' capacity to acquire grammars that comport with contemporary theories of minimalist syntax (Chomsky (1995)).²

The remainder of this study is organized as follows: after reviewing the MG formalism and prior work on modeling MGs with logic (Section 2), we present our inference procedure (Section 3) and use it to infer a set of MG lexicons from the sequence of annotated sentences listed in Table 1 (Section 4); we identify members of the inferred set of MG lexicons that are optimal with respect to cost functions inspired by the Minimum Description Length (MDL) principle (Grünwald (2007)) and the Subset principle (Berwick (1985); Wexler (1993)) and present several examples of these optimal MG lexicons, *one of which aligns with contemporary minimalist syntax, producing for each sentence in the input sequence a parse tree that matches standard syntactic analysis*. Finally,

1. Please contact the authors to obtain an implementation of the inference procedure introduced in this study.

2. For detailed presentations of minimalist syntax, see (Adger (2003); Hornstein et al. (2005); Radford (2009)).

in (Section 5) we discuss how our procedure, which takes the form of a computational model of language acquisition, may be applied to the evaluation of the Strong Minimalist Thesis.³

2. Minimalist Grammars

The Minimalist Grammar (MG) formalism, introduced in [Stabler \(1996\)](#), is a well established formal model of syntax inspired by [Chomsky \(1995\)](#). We chose to use this formalism because: (i) MGs are mildly context-sensitive ([Michaelis \(1998\)](#)) and can model cross-serial dependencies that arise in natural language ([Vijay-Shanker et al. \(1987\)](#); [Stabler \(2004\)](#)); (ii) MGs can model displacement, a basic fact of natural language that enables a phrase to be interpreted both in its final, surfaced position, as well as other positions within a syntactic structure ([Chomsky \(2013\)](#)).⁴

An MG consists of: (i) a *lexicon*, consisting of a finite set of atomic structures, referred to as *lexical items*, each of which pairs a phonetic form⁵ with a finite sequence of (syntactic) features.⁶ (ii) *merge*, a recursive structure building operation that combines two structures to produce a new structure.⁷ Each application of *merge* consumes the first feature from each of its two arguments, requiring that the two features have the same value; one of the two arguments then *projects* its feature sequence to the structure produced by merged. To parse a sentence, a set of lexical items is drawn from the lexicon and combined together, via the recursive application of *merge*, into a single structure in which all of the features have been consumed; if the ordering of the phonetic forms in the resulting structure aligns with the order of the words in the sentence being parsed, then the structure is considered to be a valid parse of the sentence. See [Figure 2](#) and [Figure 3](#) for examples of MG parses.

Finally, let us consider whether an MG may be modeled with Satisfiability Modulo Theory (SMT) ([Barrett and Tinelli \(2018\)](#)). [Rogers \(1998\)](#) established that the axioms of GB theory are expressible with Monadic Second Order logic (MSO); subsequently, [Graf \(2013\)](#) produced an MSO axiomatization for MGs⁸, and notes that over finite domains these constraints may be expressed with first order logic. As this study only considers models with finite domains, we *can* develop a finite theory of MGs with an axiomatization based in part on the MSO axiomatization of MGs

3. [Chomsky \(2008\)](#) The Strong Minimalist Thesis asserts that “*language is an optimal solution to interface conditions that FL must satisfy; that is, language is an optimal way to link sound and meaning, where these notions are given a technical sense in terms of the interface systems that enter into the use and interpretation of expressions generated by an I-language.*” See also: [Chomsky \(2001\)](#).

4. A single phrase satisfying multiple interface conditions often requires that it undergo syntactic movement to establish a discontinuous structure (i.e. a chain) with multiple local relations; by the *Principle of Last Resort*, movement is driven by morphological considerations – e.g. morphological agreement ([Chomsky \(1995\)](#)).

5. A phonetic form is either overt (e.g. a word in the sentence) or covert (i.e. unpronounced).

6. A feature has: (i) a value from a finite set of categories; (ii) a type, which is either *selector*, *selectee*, *licensor* or *licensee*, indicated by the prefix =, ~, + and – respectively; a < or > prefixed before a selector prefix indicates that the selector triggers left or right head-movement respectively. There is also a special feature, *C*, that serves to indicate the completion of a parse.

7. *Merge* applies to two logically disjoint cases: (i) *internal merge*, for the case in which one argument is a substructure of the other (i.e. they are not disjoint), requires that the consumed features for the two arguments be a *licensor* and *licensee*, with the former projecting. (ii) *external merge*, for the case in which the two arguments are disjoint, requires that the consumed features for the two arguments be a *selector* and *selectee*, with the former projecting.

8. [Graf \(2013\)](#) also shows that constraints may be encoded in an MG lexicon if and only if they are MSO expressible.

developed by Graf (2013).⁹ We will express the theory with a multi-sort quantifier-free¹⁰ first-order logic extended with the theory of uninterpreted functions (i.e. the theory of equality), allowing us to model the theory with an SMT-solver and (decidably) identify interpretations of models.

3. Inference Procedure

Our inference procedure takes the form of a computational model of language acquisition (Chomsky (1965)) consisting of: (i) an initial state, S_0 , consisting of a system of first-order logical formulae that serve as axioms for deducing the class of minimalist lexicons; (ii) the input, consisting of a sequence of n sentences, denoted I_1, I_2, \dots, I_n , each of which is annotated with syntactic relations between pairs of words in the sentence; (iii) a function, Q , that takes as input a state, S_i , and an annotated sentence, I_i , and outputs the successor state, S_{i+1} ; (iv) a function, R , that maps a state S_i to a set of MG lexicons, G_i , with the property that for each sentence I_j in the input sequence, each lexicon $L \in G_i$ can produce a parse p_j^L such that the syntactic relations in p_j^L parse match those specified in the annotation of s_j .¹¹ The procedure consumes the input sequence one annotated sentence at a time, using Q to drive the initial state, S_0 , to the final state, S_n ; the function R is then applied to S_n to produce a set of MG lexicons, G_n , that constitutes the output of the inference procedure. (See Table-1 for an example of input to the procedure)

We implemented this inference procedure by encoding an MG parser as a system of first-order, quantifier-free logical formulas that can be solved with an SMT-solver.¹² This system of formulas is composed of formulas for MG parse trees (see Section 3.1) that are connected (by way of shared symbols) to a formula for an MG lexicon (i.e. S_0); by imposing constraints on the formulas for parse trees, the set of solutions to the lexicon formula is restricted. Let us now review the role Q and R play in this.

When the inference procedure consumes an annotated sentence from the input sequence, the function Q : (1) instantiates a formula for a MG parse; (2) translates the annotations for the sentence into (logic) formulas that constrain the parse tree – e.g. predicate-argument relations and morphological agreement are translated into locality constraints¹³, and each sentence is marked as declarative or interrogative, indicating which of two pre-specified covert phonetic forms, ϵ_{CDecl} or ϵ_{CIntr} , must appear in a parse of the sentence (see Fig. 3 for an example); (3) adds these new formulas to the existing system of formulas in S_i to produce S_{i+1} .

In order to compute the set of lexicons, $G_i = R(S_i)$, we used the Z3 SMT-solver to solve for the set of lexicons satisfying the formulae in S_i .¹⁴ Note that the inferred set G_i is not enumerated; rather,

-
9. Although we use the concept of *slices*, presented in Graf (2013), our axiomatization does not utilize the first-order theory of finite trees (originally presented in Backofen et al. (1995)).
 10. The axioms in the theory must be quantifier free as the SMT-solver cannot guarantee decidability for problems involving universal quantifiers; this is established via explicit quantification.
 11. In the case of the initial state, S_0 , since there are no constraints yet imposed by the input, $R(S_0)$ will map to the set of all minimalist lexicons.
 12. All logical formulas in this study, being used to encode finite models over bounded domains, are first-order and quantifier-free; this has the benefit that these formulas are decidable.
 13. The *principle of syntactic locality* asserts that syntactic relations are established locally by merge (Sportiche et al. (2013)).
 14. Z3 is a high-performance solver for Satisfiability Modulo Theories (SMT) that can solve first-order quantifier-free multi-sort logic formulas that may combine symbols from a set of additional logics defined by a number of background theories such as empty theory (i.e. the theory of uninterpreted functions with equality). See De Moura and Bjørner (2008) for further reference.

it exists implicitly in the model produced by the SMT-solver (i.e. the solution to the system of logical formulas), and members of this set may be filtered, searched and sampled by querying this model using Z3; since the number of inferred lexicons is often exponentially large due to symmetries, we do not enumerate the entire set of lexicons; instead we use Z3 to sample lexicons from G_i .

3.1. Modeling an MG Parse Tree

We now provide an overview of a finite model of a minimalist parse tree, based closely on the MG formalism, that we have developed using a multi-sort first-order quantifier-free logic extended with the theory of equality and uninterpreted functions. The model consists of several sorts, uninterpreted functions acting over these sorts, and a set of axioms constraining these functions that every minimalist parse tree must satisfy; additionally, the syntactic relations that annotate a sentence can also be expressed as a set of axioms (first order logic formulas) that further constrain the model. An interpretation of the model (consisting of interpretations of the uninterpreted functions) is thus a minimalist parse tree that accords with the specified syntactic relations for a given sentence.

A (minimalist) parse tree¹⁵ is modeled as a labeled directed acyclic graph¹⁶, which is in turn modeled via (i) a finite sort, members of which are nodes in the graph and (ii) a set of (unary and binary) uninterpreted functions and predicates (acting over the sorts), that establish labeled edges in the graph.¹⁷ Interpretations of these functions and predicates are constrained by a set of axioms that include both: (a) an axiomatization for the MG formalism; (b) axioms that aid in expressing constraints imposed by interface conditions – e.g. axioms for structural configurations for predicate-argument structure and projection of categories.¹⁸ These axioms are derived from properties and principles of natural language syntax that are considered *universal* in so far as they apply to all natural languages (Chomsky (1995); Collins and Stabler (2016)). Let us now review aspects of linguistic theory from which these axioms are derived and discuss how these axioms constrain the model of the MG parse tree.

In accordance with the theory of Bare Phrase Structure (BPS) (Chomsky (1995)), one of the uninterpreted functions is a binary function (over the nodes in the graph) that is constrained by axioms that model the recursive structure building operation, *Merge* (Chomsky (1995); Collins and Stabler (2016)); another unary functions models the chains produced by the movement of phrases within the parse tree.¹⁹ Each node in the parse tree has a *head*, which is one of the leaf nodes (which correspond to lexical items) in the parse tree²⁰; this mapping is established by a unary uninterpreted function. Each node in the parse tree is labeled with a category. Categories are interpretable properties of lexical items that can project – the category associated with a given

15. This tree corresponds to the derivation tree in the MG formalism.

16. Each node in this graph corresponds to a node in the parse tree and the graph is constrained so as to have a single element with no out-going edges (which corresponds to the root of the parse tree); nodes with no incoming edges correspond to atomic syntactic structures.

17. E.g. an uninterpreted binary predicate models dominance relations between nodes in the graph, and the transitive closure of this predicate establishes a binary tree in accordance with the Binary Branching Hypothesis (Radford (2009)).

18. Along with the finite sort that constitutes the nodes of the derivation tree, a number of additional finite sorts and functions mapping to and from them are employed to represent phonetic forms, features, categories, etc.

19. An uninterpreted function for modeling head movement is also included, with relevant axioms in accordance with the Head-Movement constraint as given in (Baker (1988); Hale (1993); Stabler (2001))

20. See Radford (2009) for a discussion of the *Headedness Principle*, according to which “every nonterminal node in a syntactic structure is a projection of a head word.”

I_i	Sentence	Syntactic Relations
I_1	“John has eaten pizza.”	agree(John, has), arg(John, eaten), arg(pizza, eaten)
I_2	“Has Sally eaten pizza?”	agree(Sally, has), arg(Sally, eaten), arg(pizza, eaten)
I_3	“What has John eaten?”	agree(John, has), arg(John, eaten), arg(What, eaten)
I_4	“Who has eaten pizza?”	agree(Who, has), arg(Who, eaten), arg(pizza, eaten)
I_5	“Sally was eating pizza.”	agree(Sally, was), arg(Sally, eating), arg(pizza, eating)
I_6	“Was John eating pizza?”	agree(John, was), arg(John, eating), arg(pizza, eating)
I_7	“What was Sally eating?”	agree(Sally, was), arg(Sally, eating), arg(What, eating)
I_8	“Who was eating pizza?”	agree(Who, was), arg(Who, eating), arg(pizza, eating)
I_9	“Pizza was eaten.”	agree(pizza, was), arg(pizza, eaten)
I_{10}	“Was pizza eaten?”	agree(pizza, was), arg(pizza, eaten)
I_{11}	“What was eaten?”	agree(What, was), arg(What, eaten)

Table 1: Input sequence of annotated sentences. The annotation of each sentence includes the syntactic relations listed for morphological agreement (indicated by *agree*) and predicate-argument structure (indicated by *arg*); the type of the sentence – i.e. either declarative or interrogative – is also annotated on each sentence (but not listed here). The sentences listed here include passive constructions (I_9, I_{10}, I_{11}), yes/no-questions (I_2, I_6, I_{10}) and wh-questions ($I_3, I_4, I_7, I_8, I_{11}$).

phrase is the category associated with the head of that phrase. An additional finite sort encodes the universal functional and lexical categories $\{C_{Declarative}, C_{Question}, T, v, V\}$ and $\{P, D, N\}$ (Adger and Svenonius (2011)) and additional functions and axioms encode the two extended projections C-T-V and P-D-N (Grimshaw (2005); Adger (2003)) that constrain what structural configurations the functional categories may be arranged in within a derivation that converges.

Finally, we consider the axioms imposed on a given sentence by: (a) the linear ordering of the words in a sentence – *linearization* is modeled by a conjunction of axioms for *lifting the derived tree from the derivation tree* as presented in Graf (2013)²¹ and Kayne’s Linear Correspondence Axioms (Kayne (1994)); (b) the syntactic relations annotating a sentence – these are each translated into axioms for either morphological agreement or predicate-argument structure, the latter in accordance with the theory of argument structure in (Hale (1993)).²²

4. Experiment

We used our implementation of the procedure presented in Section 3 to infer a set of minimalist lexicons, denoted here as G^* , from an input sequence with eleven sentences (listed in Table-1), each annotated with predicate-argument relations as well as morphological agreement.²³ We validated the lexicons sampled from G^* by using an agenda-based MG parser (Harkema (2001)) to verify that the lexicon can be used to parse each sentence in the input sequence.

Manual inspection of lexicons sampled from G^* revealed that many of them had a large number of lexical items and produced parses that do not resemble those found in contemporary theories of syntax – see Lexicon-A in Figure-1 for an example.

21. See also Collins and Stabler (2016).

22. This theory requires that the lexical heads associated with a predicate and its arguments must enter into particular structural configurations within a derivation. (See also Hale and Keyser (2002).

23. We bounded the acquisition model with the following parameters: a parse may have up to 3 instances of phrasal movement and up to one instance of head movement; lexical items may have at most 3 features.

Lexicon-A	Lexicon-B	Lexicon-C
eaten::<=x2,+1,~x1	eaten::=x2,~x2	eaten::=x3,~x4
eaten::=x2,+1,~x2	eaten::=x2,+r,~x2	eaten::=x0,=x0,~x1
eating::<=x2,+1,~x1	eating::=x2,~x2	eating::=x0,=x0,~x4
eating::=x2,+1,~x2	has::=x1,+1,~x2	has::=x1,+1,~x2
has::~~x0,-1	has::=x2,+1,~x2	john::~~x0,-1
has::=x2,+r,~x1	has::~~x2,-r,-r	pizza::~~x0
john::=x0,~x2	john::=x2,+r,~x2	pizza::~~x3,-1
john::~~x2,-1,-r	pizza::=x2,+r,~x2	sally::~~x0,-1
pizza::=x2,~x2	sally::=x2,+r,~x2	was::=x4,+1,~x2
pizza::=x2,+1,~x2	was::=x2,+1,~x2	what::~~x0,-r
sally::~~x2,-1,-r	was::=x2,+1,~x2	what::~~x3,-1,-r
sally::=x2,+1,~x2	what::~~x2,-r,-1	who::~~x0,-1,-r
was::~~x2,-1,-1	who::=x2,+r,~x0	$\epsilon_{Cdecl}::=x2,C$
was::=x2,+r,~x1	$\epsilon_{Cdecl}::<=x2,C$	$\epsilon_{Cintr}::<=x2,C$
what::=x1,~x1	$\epsilon_{Cintr}::=x2,C$	$\epsilon_{Cintr}::<=x2,+r,C$
what::=x2,+1,~x1	$\epsilon_{Cintr}::<=x0,C$	
who::~~x2,-1,-r		
$\epsilon_{Cdecl}::<=x2,C$		
$\epsilon_{Cdecl}::=x1,C$		
$\epsilon_{Cintr}::=x1,C$		
$\epsilon_{Cintr}::<=x1,C$		

Figure 1: Examples of inferred lexicons that satisfy the conditions imposed by the input sequence in Table-1. Each lexical item consists of a pairing of a phonetic form and a sequence of syntactic features, separated by a double-colon. The phonetic forms ϵ_{Cdecl} and ϵ_{Cintr} are covert (unpronounced); their presence in a minimalist parse tree indicates whether the parse is of a declarative or interrogative sentence, respectively.

We filtered out lexicons such as these by using Z3 to identify lexicons in G^* that were *optimal* with respect to a cost function that penalizes a lexicon for the number of lexical entries it has.²⁴ This produced a subset of G^* in which every lexicon had exactly 15 lexical items, the minimal number of lexical items required for a lexicon to be able to produce parses that accord with the specified input sequence. See Lexicon-B in Figure-1 for an example of a lexicon in this subset; see Figure-3 for an example of a parse produced by Lexicon-B.

We manually inspected this subset of G^* and found that most of the lexicons produced parses with many instances of internal merge that could have been eliminated without any side-effects, and that these parses that did not accord with contemporary theories of syntax.

Finally, we further refined this subset of G^* by using Z3 to identify lexicons that were *optimal* with respect to two additional cost functions:

1. (minimize) the total number of selectional and licensing features in the lexicon and the parses; this cost function rewards reduction in the total size of both the lexicon and the derivations,²⁵

24. We did this by encoding this cost function as a logical formula, adding it to the SMT-solver after running the inference procedure, and then re-solving; the resulting set of (inferred) minimalist grammars are optimal with respect to the specified cost functions.

25. This cost function is based on the MDL principle (Grünwald (2007)) as applied to MGs in (Stabler (1998)).

2. (maximize) the number of *distinct* selectional features in the lexicon; this cost function rewards lexicons that are less inclusive (i.e. they are less likely to overgenerate).²⁶

This produced a subset of G^* in which each lexicon had exactly: 15 lexical items; 33 features in the lexicon (not including the special feature C); 125 features in the parses; at least 4 distinct selectional features. See Lexicon-C in Figure-1 for a representative member of this subset. We found that the lexicons in this subset were all of the same form – i.e. they are only differentiated by permutations of the feature values and other symmetries in the model – *and that these lexicons produced parses that agreed with those prescribed by contemporary minimalist theories of syntax.* (as presented in [Hornstein et al. \(2005\)](#), [Adger \(2003\)](#), and [Radford \(1997\)](#).) See Figure-2 for a parse produced by Lexicon-C that demonstrates several of the syntactic phenomenon that Lexicon-C models correctly (i.e. as prescribed by minimalist theories of syntax) while respecting the syntactic relations prescribed in sentence I_7 of Table-1.

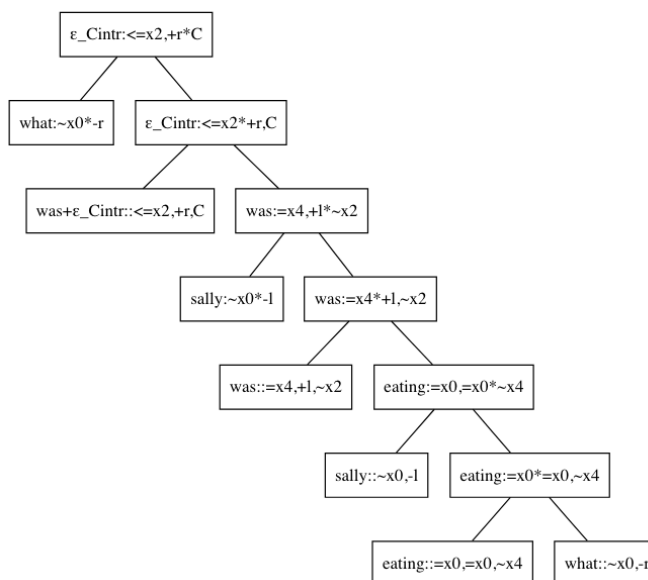


Figure 2: An MG parse for the sentence “What was Sally eating?” (see I_7 in Table-1 for annotations) derived from Lexicon-C in Figure-1. The parse is assembled in a bottom-up manner via merge: first “eating” merges with “what” and then with “Sally”, thus establishing (via locality) predicate-argument relations; once the resulting structure then merges with “was”, “sally” then undergoes subject-raising by (internally) merging with “was”, thus establishing morphological agreement between “sally” and “was”. Next the head of the auxiliary verb “was” undergoes head movement to merge with the covert form, ϵ_{Cintr} , which indicates that the sentence is an interrogative. Finally, “what” undergoes wh-fronting by (internally) merging with ϵ_{Cintr} . Wh-fronting and Subject-raising are triggered by different licenser features, the former by $+r$ and the latter by $+l$. The feature sequences displayed in internal nodes have an asterisk separating features that have already been consumed (on the left) from those that have not (on the right).

26. This cost function is based on the Subset Principle ([Berwick \(1985\)](#)), which asserts that a language learner will always choose the least inclusive grammar available at each stage of acquisition; the adaption of this principle is a logical necessity if one assumes that the learner does not make use of (indirect) negative evidence. See also [Yang \(2015, 2016\)](#).

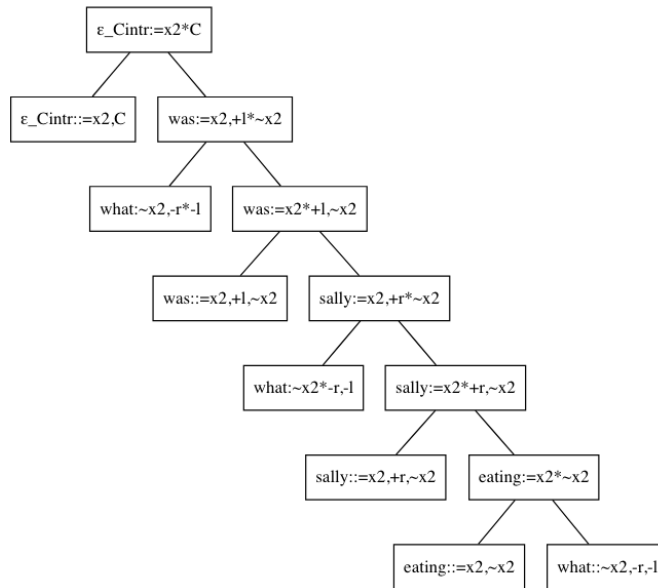


Figure 3: An MG parse for the sentence “What was Sally eating?” derived from Lexicon-B in Figure-1. Although this parse satisfies the constraints imposed by the annotations listed for I_7 in Table-1, it does not accord with the parse prescribed by contemporary theories of syntax – e.g. nowhere in the parse does the auxiliary verb “was” establish a local relation with the main predicate “eating.”

5. Conclusion

In this study we have (i) proposed and implemented a procedure for inferring MGs and (ii) used this procedure to infer an MG that closely aligns with contemporary theories of syntax, thus demonstrating how linguistically-relevant MGs may be identified within the inferred set of MGs by optimizing cost functions derived from methods of inductive inference that are relevant to cognitively-faithful models of language acquisition. We observe that by enabling and disabling axioms in our model, we can carry out experiments to determine which are redundant, and *thereby gain insight into whether the linguistic principles, from which the axioms of the system are largely derived, are justified or can be discarded*, thus aiding in the evaluation of the Strong Minimalist Thesis.

Going forward, we plan to: incorporate phase theory (Chomsky (2001, 2008)) into our model of a minimalist parse tree, following the approach taken by Chesi (2007); examine the over-generations produced by the MGs inferred by our procedure and understand how these over-generations relate to the cost functions used by our procedure for identifying optimal grammars; investigate the potential for this procedure to be used for producing MG treebanks²⁷, which may aid treebank based parsing strategies, by extracting sets of (partially) annotated sentences (that may be used as input to the inference procedure) from treebanks such as PropBank (Kingsbury and Palmer (2002)) or the UD treebanks (Nivre et al. (2016)).

27. See Torr (2018) for an alternative approach to developing large-scale MG treebanks.

Acknowledgments

The author would like to thank Robert C. Berwick, Sandiway Fong, Beracah Yankama, and Norbert Hornstein for their suggestions, feedback, and inspiration. Additionally, the author is very grateful for the financial support provided by Moody's Investor Services.

References

- David Adger. *Core syntax: A minimalist approach*, volume 33. Oxford University Press Oxford, 2003.
- David Adger and Peter Svenonius. Features in minimalist syntax. *The Oxford handbook of linguistic minimalism*, pages 27–51, 2011.
- Rolf Backofen, James Rogers, and Krishnamurti Vijay-Shanker. A first-order axiomatization of the theory of finite trees. *Journal of Logic, Language and Information*, 4(1):5–39, 1995.
- Mark C. Baker. *Incorporation: A theory of grammatical function changing*. University of Chicago Press, 1988.
- Clark Barrett and Cesare Tinelli. Satisfiability modulo theories. In *Handbook of Model Checking*, pages 305–343. Springer, 2018.
- Robert C. Berwick. *The acquisition of syntactic knowledge*. MIT press, 1985.
- Cristian Cadar and Koushik Sen. Symbolic execution for software testing: three decades later. *Commun. ACM*, 56(2):82–90, 2013.
- Cristiano Chesi. An introduction to phase-based minimalist grammars: why move is top-down from left-to-right. *Studies in linguistics*, 2007.
- Noam Chomsky. *Aspects of the theory of syntax*. MIT Press, 1965.
- Noam Chomsky. *The Minimalist Program*. MIT Press, 1995.
- Noam Chomsky. Derivation by phase. In Michael Kenstowicz, editor, *Ken Hale: A life in language*, pages 1–52. MIT press, 2001.
- Noam Chomsky. On phases. *Current Studies in Linguistics Series*, 45:133, 2008.
- Noam Chomsky. Problems of projection. *Lingua*, 130:33–49, 2013.
- Chris Collins and Edward Stabler. A formalization of minimalist syntax. *Syntax*, 19(1):43–78, 2016.
- Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'08/ETAPS'08*, pages 337–340. Springer-Verlag, 2008.

- Leonardo De Moura and Nikolaj Bjørner. Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9):69–77, 2011.
- Thomas Graf. *Local and transderivational constraints in syntax and semantics*. PhD thesis, University of California at Los Angeles, 2013.
- Jane B Grimshaw. *Words and structure*. Center for the Study of Language and Information, Stanford University, 2005.
- Peter D Grünwald. *The minimum description length principle*. MIT press, 2007.
- Kenneth Hale. On argument structure and the lexical expression of syntactic relations. In Ken Hale and Samuel J. Keyser, editors, *The view from Building 20: Essays in linguistics in honor of Sylvain Bromberger*. MIT Press, 1993.
- Kenneth Locke Hale and Samuel Jay Keyser. *Prolegomenon to a theory of argument structure*, volume 39. MIT press, 2002.
- Hendrik Harkema. *Parsing minimalist languages*. Citeseer, 2001.
- Norbert Hornstein, Jairo Nunes, and Kleantes K Grohmann. *Understanding minimalism*. Cambridge University Press, 2005.
- Richard S Kayne. *The antisymmetry of syntax*. Number 25. MIT Press, 1994.
- Paul Kingsbury and Martha Palmer. From treebank to propbank. In *LREC*, pages 1989–1993. Citeseer, 2002.
- Jens Michaelis. Derivational minimalism is mildly context-sensitive. In *LACL*, volume 98, pages 179–198. Springer, 1998.
- Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *LREC*, 2016.
- Fernando C. N. Pereira and David H. D. Warren. Parsing as deduction. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, ACL '83, pages 137–144. Association for Computational Linguistics, 1983.
- Andrew Radford. *Syntactic theory and the structure of English: A minimalist approach*. Cambridge University Press, 1997.
- Andrew Radford. *An introduction to English sentence structure*. Cambridge university press, 2009.
- Manny Rayner, Åsa Hugosson, and Göran Hager. Using a logic grammar to learn a lexicon. In *Proceedings of the 12th conference on Computational linguistics-Volume 2*, pages 524–529. Association for Computational Linguistics, 1988.
- James Rogers. *A descriptive approach to language-theoretic complexity*. CSLI Publications, 1998.
- Dominique Sportiche, Hilda Koopman, and Edward Stabler. *An introduction to syntactic analysis and theory*. John Wiley & Sons, 2013.

- Edward Stabler. Derivational minimalism. In *International Conference on Logical Aspects of Computational Linguistics*, pages 68–95. Springer, 1996.
- Edward P Stabler. *The Logical Approach to Syntax*. MIT Press, 1993.
- Edward P. Stabler. Acquiring languages with movement. *Syntax*, 1(1):72–97, 1998.
- Edward P Stabler. Recognizing head movement. In *International Conference on Logical Aspects of Computational Linguistics*, pages 245–260. Springer, 2001.
- Edward P Stabler. Varieties of crossing dependencies: structure dependence and mild context sensitivity. *Cognitive Science*, 28(5):699–720, 2004.
- John Torr. Constraining mgbank: Agreement, l-selection and supertagging in minimalist grammars. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 590–600. Association for Computational Linguistics, 2018.
- K. Vijay-Shanker, David J. Weir, and Aravind K. Joshi. Characterizing structural descriptions produced by various grammatical formalisms. In *Proceedings of the 25th Annual Meeting on Association for Computational Linguistics*, ACL ’87, pages 104–111. Association for Computational Linguistics, 1987.
- Kenneth Wexler. The subset principle is an intensional principle. In *Knowledge and language*, pages 217–239. Springer, 1993.
- Charles Yang. Negative knowledge from positive evidence. *Language*, 91(4):938–953, 2015.
- Charles Yang. *The price of linguistic productivity: How children learn to break the rules of language*. MIT Press, 2016.