Incremental Acquisition of a Minimalist Grammar using an SMT-Solver

Sagar Indurkhya
MIT
32 Vassar St.
Cambridge, MA 02139
indurks@mit.edu

A central question in cognitive linguistics is how children everywhere can readily acquire *Knowledge of Language* (KoL) from (restrictive) Primary Linguistic Data (PLD) (Chomsky, 1986; Berwick et al., 2011; Chomsky, 2013). This study addresses this question by introducing a novel procedure, implemented as a working computer program, that uses an interactive theorem prover to incrementally infer a Minimalist Grammar (MG) (Stabler, 1996). The procedure, which is inspired by (Rayner et al., 1988) and builds on earlier work by (Indurkhya, 2020), takes the form of a computational model of child language acquisition (Berwick, 1985; Chomsky, 1965). The procedure takes as input a sequence of paired interface conditions - i.e. each entry is a Phonological Form (PF), encoding a sentence, paired with a Logical Form (LF), which encodes thematic roles for each predicate as well as agreement relations; the input sequence, which corresponds to the PLD that a child is exposed to, is organized into a sequence of batches that the procedure consumes incrementally. The procedure outputs an MG lexicon, consisting of a set of (word, feature-sequence) pairings, that yields, for each entry in the input sequence, a minimalist derivation that satisfies the listed interface conditions; the output MG lexicon corresponds to the KoL that the child acquires from processing the PLD.

The procedure, which models a child language learner, operates as follows. The initial state of the the learner is an empty MG lexicon. The procedure incrementally constructs an MG lexicon: at each step the procedure takes as input a batch of the PLD and the lexicon that constitutes the current state of the learner, and then it augments the input lexicon with the minimal set of additional lexical entries needed to ensure that the augmented lexicon will yield, for each entry in the batch of PLD, a minimalist derivation that satisfies the listed interface conditions.[1] When processing a batch of the PLD, the learner first constructs a set of logical formulae, expressed using the logic of Satisfiability Modulo Theories (SMT) (De Moura and Bjørner, 2011), that encodes: (i) an SMT-model of an MG lexicon that is required to have at least the lexical entries in the input lexicon; (ii) for each entry in the batch of PLD, an SMT-model of an MG derivation that must be derivable from the lexicon and that must satisfy the interface conditions listed for that entry.[2] The procedure then employs the Z3 SMT-solver (De Moura and Bjørner, 2008) to identify a solution to this set of SMT formulae that corresponds to the smallest[3] lexicon, and from this solution the "augmented" lexicon, which is the new state of the learner, is automatically recovered.[4] The final output of the procedure – i.e. the MG lexicon yielded after consuming the full PLD – corresponds to the KoL that the learner has acquired. Importantly, at a given step of the procedure, the size of the SMT-model of the lexicon is constrained by the size of the input lexicon (and a small, fixed, number of lexical entries that may be added), and the number of SMT-models of derivations is constrained by the size of the PLD-batch - *thus, the procedure can iteratively consume a large PLD without blowing up the size of the constructed SMT-models, thereby avoiding computational intractibility.*

[1]Each iteration of this process corresponds to an application of the instantaneous MG acquisition procedure introduced in (Indurkhya, 2020) and detailed in §3.2 of (Indurkhya, 2021a).

[2]The SMT-model of the lexicon is linked to each SMT-model of a derivation via common free-variables. See Ch. 2 of (Indurkhya, 2021a) for a complete presentations of these SMT-models.

[3]As measured by (firstly) the number of distinct feature sequences that appear in the lexicon, and (secondly) the total number of features that appear in the lexicon. Unlike (Indurkhya, 2020), here the acquisition procedure is restricted to work with a single selectional feature, x_0, which has the benefit of reducing the size of the SMT model, but yields a lexicon that is underconstrained w.r.t. c-selection; see (Indurkhya, 2021b) for a discussion of how model based collaborative filtering could be used to constrain which arguments a predicate can select within a derivation.

[4]This augmented lexicon is a superset of the input lexicon.

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 212-216.
Held on-line February 7-9, 2022

Batch	I_i	Interface	Interface Conditions
2	I_{29}	PF	john/N has asked/V whether pizza/N was eaten/V.
		LF	θ_{asked}[s: john, o: whether pizza was eaten], Agr_{has}[s: john], θ_{eaten}[o: pizza], Agr_{was}[s: pizza]
	I_{30}	PF	mary/N was told/V that john/N has eaten/V pizza/N.
		LF	θ_{told}[o: that john has eaten pizza, i: mary], Agr_{was}[s: mary], θ_{eaten}[s: john, o: pizza], Agr_{has}[s: john]
	I_{31}	PF	mary/N has told/V john/N that icecream/N was eaten/V.
		LF	θ_{told}[s: mary, o: that icecream was eaten, i: john], Agr_{has}[s: mary], θ_{eaten}[o: icecream], Agr_{was}[s: icecream]
	I_{32}	PF	mary/N has asked/V john/N whether she/N was eating/V pizza/N.
		LF	θ_{asked}[s: mary, o: whether she was eating pizza, i: john], Agr_{has}[s: mary], θ_{eating}[s: she, o: pizza], Agr_{was}[s: she]
	I_{33}	PF	who has mary/N told/V that she/N was eating/V icecream/N?
		LF	θ_{told}[s: mary, o: that she was eating icecream, i: who], Agr_{has}[s: mary], θ_{eating}[s: she, o: icecream], Agr_{was}[s: she]
	I_{34}	PF	who was asked/V whether mary/N has given/V john/N money/N?
		LF	θ_{asked}[o: whether mary has given john money, i: who], Agr_{was}[s: who], θ_{given}[s: mary, o: money, i: john], Agr_{has}[s: mary]
3	I_{35}	PF	who has told/V john/N everything/N that mary/N was asked/V?
		LF	θ_{told}[s: who, o: everything that mary was asked, i: john], Agr_{has}[s: who], θ_{asked}[o: everything, i: mary], Agr_{was}[s: mary]
	I_{36}	PF	was someone/N given/V everything/N that she/N has eaten/V?
		LF	θ_{given}[o: everything that she has eaten, i: someone], Agr_{was}[s: someone], θ_{eaten}[s: she, o: everything], Agr_{has}[s: she]
4	I_{37}	PF	mary/N has seen/V everyone/N who john/N was eating/V.
		LF	θ_{seen}[s: mary, o: everyone who john was eating], Agr_{has}[s: mary], θ_{eating}[s: john, o: everyone], Agr_{was}[s: john]
	I_{38}	PF	john/N has seen/V someone/N who was eating/V icecream/N.
		LF	θ_{seen}[s: john, o: someone who was eating icecream], Agr_{has}[s: john], θ_{eating}[s: someone, o: icecream], Agr_{was}[s: someone]
	I_{39}	PF	john/N has seen/V someone/N who was eaten/V.
		LF	θ_{seen}[s: john, o: someone who was eaten], Agr_{has}[s: john], θ_{eaten}[o: someone], Agr_{was}[s: someone]

Figure 1: The acquisition procedure incrementally inferred the grammar listed in Fig. 2 after (successively) consuming four batches of primary linguistic data (PLD), the latter three of which are shown here (see Table 3.2 in (Indurkhya, 2021a) for the first batch). Each of the input sentences listed here has one degree (level) of embedding. The second batch consist of sentences in which the embedded clause is a declarative (e.g. I_{31}) or an interrogative (e.g. I_{34}). The third and fourth batch consists of sentences with an embedded (restrictive) relative clause. Notably, in the case of LF interface conditions that mark an embedded clause as an argument, the tokens making up the embedded clause are interpreted as a multi-set of phonological forms - e.g. the LF interface conditions for I_{35} indicate that the phrase to be formed from the multi-set of phonological forms { everything, that, mary, was, asked } will serve as an internal argument of the (lexical) verb "told". Hence, the LF interface conditions *do not explicitly encode any information about the linear ordering of the phonological forms that form the sentence, and only serve to constrain the hierarchical relations that establish predicate-argument structure.*

ID	Category	Features	what	who	***that***	***whether***	has	was	***she***	everything	someone	icecream	money	mary	pizza	john	nothing	boy	story	given	told	asked	known	***seen***	asking	eating	eaten	to	***everyone***	sleeping	slept	the	a	ϵ
$\mathfrak{L}_1$	V	$= x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	×	×	×	×	×	·	·	·	·	·	·	·
$\mathfrak{L}_2$	V	$= x_0, = x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	×	×	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_3$	$C_{decl.}$	$= x_0, C$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×
$\mathfrak{L}_4$	$C_{ques.}$	$<= x_0, +z, C$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×
$\mathfrak{L}_5$	v	$<= x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×
$\mathfrak{L}_6$	$C_{ques.}$	$<= x_0, C$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×
$\mathfrak{L}_7$	v	$<= x_0, = x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×
$\mathfrak{L}_8$	P	$= x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	·	·	·	·	·	·
$\mathfrak{L}_9$	D	$= x_0, \sim x_0, -l$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	×	·
$\mathfrak{L}_{10}$	D	$= x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	×	·
$\mathfrak{L}_{11}$	D	$\sim x_0, -z$	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{12}$	D	$\sim x_0, -l, -z$	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{13}$	T	$= x_0, +l, \sim x_0$	·	·	·	·	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{14}$	V	$\sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	×	·	·	·
$\mathfrak{L}_{15}$	N	$\sim x_0, -l$	·	·	·	·	·	·	×	×	×	×	×	×	×	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{16}$	N	$\sim x_0$	·	·	·	·	·	·	·	×	×	×	×	×	×	×	×	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{17}$	$C_{decl.}$	$= x_0, \sim x_0$	·	·	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{18}$	$C_{decl.}$	$= x_0, +z, \sim x_0$	·	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{19}$	N	$\sim x_0, -z$	·	·	·	·	·	·	·	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	·	·	·	·	·
$\mathfrak{L}_{20}$	N	$\sim x_0, -l, -z$	·	·	·	·	·	·	·	·	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·

Figure 2: A factored representation of an MG lexicon inferred by the acquisition procedure - an X indicates that a lexical entry pairing a feature sequence with phonological form. Bold horizontal-lines dividing the feature-sequences indicate which of the four (successive) PLD batches was processed when those feature-sequences were added; notably, the lexicon was augmented with only four feature-sequences ($\mathfrak{L}_{17}$ - $\mathfrak{L}_{20}$) to handle the embedded clauses found in PLD batches 2-4. Bolded phonological forms appeared after the first PLD batch - e.g. "she" first appears in the second PLD batch, and is paired with $\mathfrak{L}_{15}$ (that was added upon processing the first PLD batch).

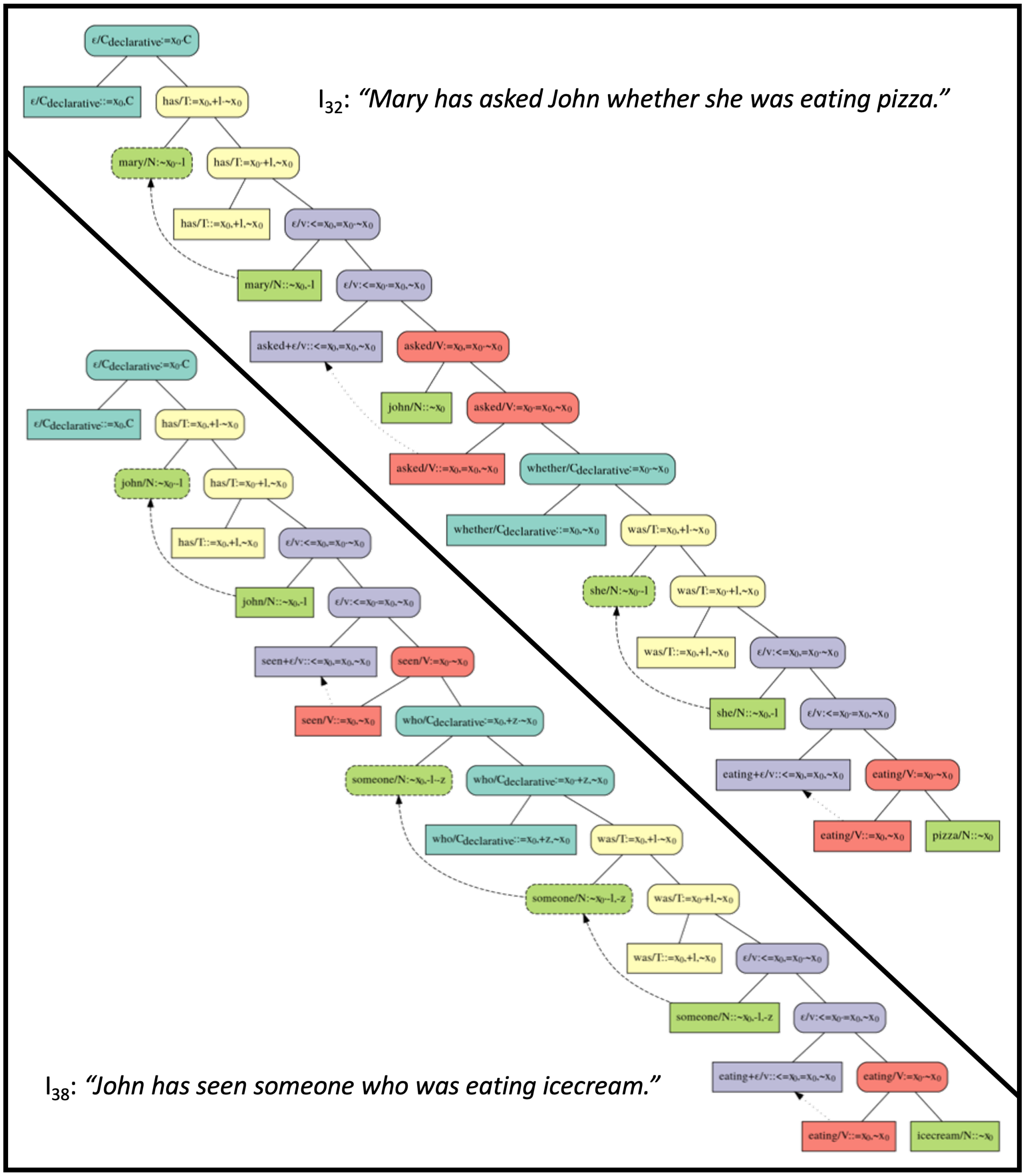

Figure 3: Presentations of two MG derivations identified by the acquisition procedure that satisfy the interface conditions listed in I_{32} and I_{38} respectively and that may be yielded by the lexicon listed in Fig. 2. The derivation for I_{32}, which derives a sentence with an embedded question, employs feature-sequences $\{\mathfrak{L}_1, \mathfrak{L}_2, \mathfrak{L}_3, \mathfrak{L}_7, \mathfrak{L}_{13}, \mathfrak{L}_{15}, \mathfrak{L}_{16}, \mathfrak{L}_{17}\}$, and the derivation for I_{38}, which derives a sentence with an embedded restrictive relative clause, employs feature-sequences $\{\mathfrak{L}_1, \mathfrak{L}_3, \mathfrak{L}_7, \mathfrak{L}_{13}, \mathfrak{L}_{15}, \mathfrak{L}_{16}, \mathfrak{L}_{18}, \mathfrak{L}_{20}\}$. The leaf nodes (indicated by absence of rounded corners) are lexical items selected from the lexicon. The derivation is assembled in a bottom-up manner via repeated applications of the structure-building operation *merge*. The feature sequences displayed in non-leaf nodes (indicated by rounded corners) have a dot, · , that separates those features that have already been consumed (on the left) from those that have not (on the right) - see (Stabler, 2001) for details the MG feature system. Nodes with the same head have the same color. Head-movement is indicated by the dotted-arrows, and phrasal movement is indicated by the dashed arrows. Note that as the system is only provided with two types of complementizers, $C_{decl.}$ and $C_{ques.}$, the system re-uses the declarative complementizer sub-category for both types of embedded clauses; expanding the set of (sub-)categories available to the system is one possible avenue of future improvement to the system.

Base Case (Degree-0): *"A boy has told someone the story."*

$$\{\frac{\epsilon_C}{\mathfrak{L}_3}\{\frac{\mathit{has}}{\mathfrak{L}_{13}}\{\{\frac{a}{\mathfrak{L}_9}\frac{\mathit{boy}}{\mathfrak{L}_{16}}\}\{\frac{\epsilon_v}{\mathfrak{L}_7}\{\frac{\mathit{someone}}{\mathfrak{L}_{16}}\{\frac{\mathit{told}}{\mathfrak{L}_2}\{\boxed{\frac{\mathit{the}}{\mathfrak{L}_{10}}\frac{\mathit{story}}{\mathfrak{L}_{16}}}\}\}\}\}\}\}\}$$

Rule: *"<u>the story</u>"* → *"that a boy has told someone <u>the story</u>"*

$$\{\boxed{\frac{\mathit{the}}{\mathfrak{L}_{10}}\frac{\mathit{story}}{\mathfrak{L}_{16}}}\} \rightarrow \{\frac{\mathit{that}}{\mathfrak{L}_{17}}\{\frac{\mathit{has}}{\mathfrak{L}_{13}}\{\{\frac{a}{\mathfrak{L}_9}\frac{\mathit{boy}}{\mathfrak{L}_{16}}\}\{\frac{\epsilon_v}{\mathfrak{L}_7}\{\frac{\mathit{someone}}{\mathfrak{L}_{16}}\{\frac{\mathit{told}}{\mathfrak{L}_2}\{\boxed{\frac{\mathit{the}}{\mathfrak{L}_{10}}\frac{\mathit{story}}{\mathfrak{L}_{16}}}\}\}\}\}\}\}\}$$

Degree-1: *"A boy has told someone [that a boy has told someone <u>the story</u>]."*

$$\{\frac{\epsilon_C}{\mathfrak{L}_3}\{\frac{\mathit{has}}{\mathfrak{L}_{13}}\{\{\frac{a}{\mathfrak{L}_9}\frac{\mathit{boy}}{\mathfrak{L}_{16}}\}\{\frac{\epsilon_v}{\mathfrak{L}_7}\{\frac{\mathit{someone}}{\mathfrak{L}_{16}}\{\frac{\mathit{told}}{\mathfrak{L}_2}\{\frac{\mathit{that}}{\mathfrak{L}_{17}}\{\frac{\mathit{has}}{\mathfrak{L}_{13}}\{\{\frac{a}{\mathfrak{L}_9}\frac{\mathit{boy}}{\mathfrak{L}_{16}}\}\{\frac{\epsilon_v}{\mathfrak{L}_7}\{\frac{\mathit{someone}}{\mathfrak{L}_{16}}\{\frac{\mathit{told}}{\mathfrak{L}_2}\{\boxed{\frac{\mathit{the}}{\mathfrak{L}_{10}}\frac{\mathit{story}}{\mathfrak{L}_{16}}}\}\}\}\}\}\}\}\}\}\}\}\}\}$$

Degree-2: *"A boy has told someone [that a boy has told someone [that a boy has told someone <u>the story</u>]]."*

$$\{\frac{\epsilon_C}{\mathfrak{L}_3}\{\frac{\mathit{has}}{\mathfrak{L}_{13}}\{\{\frac{a}{\mathfrak{L}_9}\frac{\mathit{boy}}{\mathfrak{L}_{16}}\}\{\frac{\epsilon_v}{\mathfrak{L}_7}\{\frac{\mathit{someone}}{\mathfrak{L}_{16}}\{\frac{\mathit{told}}{\mathfrak{L}_2}\{\frac{\mathit{that}}{\mathfrak{L}_{17}}\{\frac{\mathit{has}}{\mathfrak{L}_{13}}\{\{\frac{a}{\mathfrak{L}_9}\frac{\mathit{boy}}{\mathfrak{L}_{16}}\}\{\frac{\epsilon_v}{\mathfrak{L}_7}\{\frac{\mathit{someone}}{\mathfrak{L}_{16}}\{\frac{\mathit{told}}{\mathfrak{L}_2}\{\frac{\mathit{that}}{\mathfrak{L}_{17}}\{\frac{\mathit{has}}{\mathfrak{L}_{13}}\{\{\frac{a}{\mathfrak{L}_9}\frac{\mathit{boy}}{\mathfrak{L}_{16}}\}\{\frac{\epsilon_v}{\mathfrak{L}_7}\{\frac{\mathit{someone}}{\mathfrak{L}_{16}}\{\frac{\mathit{told}}{\mathfrak{L}_2}\{\boxed{\frac{\mathit{the}}{\mathfrak{L}_{10}}\frac{\mathit{story}}{\mathfrak{L}_{16}}}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}\}$$

Figure 4: A demonstration of how, by repeated application of a substitution-rule to a base-case derivation, the lexicon (listed in Fig. 2) that the acquisition procedured inferred from a PLD restricted to sentences with degree-0/1 embedding can yield a sentence with degree-n embedding for any $n \geq 0$. Note that these derivations only show what external merge operations would be applied, with internal merge assumed to immediately and automatically be applied whenever possible in the course of a derivation.

We demonstrate the capabilities of the acquisition procedure by using it to infer an MG lexicon from a PLD consisting of 39 simple sentences that were divided into four consecutive batches (having 28, 6, 2 and 3 entries respectively), with the first batch having sentences without any embedding, and the remaining batches (presented in Table 1) consisting of sentences with at most one degree of embedding (i.e. embedded declaratives or relative clauses). The procedure outputs an MG lexicon (see Fig. 2) that yields derivations for declaratives, yes/no-questions, and wh-questions in both active and passive voice; these derivations involve various forms of syntactic movement including wh-raising, subject-raising, T-to-C head-movement and V-to-v head-movement; the lexicon also includes entries for covert complementizers and light-verbs. The inferred lexicon aligns with contemporary theories of minimalist syntax[5] in so far as: (i) the lexicon yields the prescribed derivations for a variety of syntactic structures, utilizing syntactic movement (including head-movement) and covert lexical items as needed (see Fig. 3 for examples); (ii) expressions with related interpretations are assigned derivations systematically related by structural transformations. Furthermore, this lexicon can generate a countably infinite set of minimalist derivations, including derivations with n-levels of embedding for any $n \geq 0$, thereby generalizing beyond the input PLD (see Fig. 4 for more details).[6] Notably, the procedure does this without being provided a treebank of minimalist derivations that serve as examples of what the acquired lexicon should be able to yield, and to that end, the procedure constitutes a novel scheme for unsupervised inference of MGs.

The acquisition procedure demonstrates how an SMT-solver can aid in the study of linguistic theory: the solver enables us to separate out the questions of what KoL the learner acquires and how the learner acquires it – i.e. we can setup computational experiments in which we focus on specifying the learner's initial state and the conditions that the learner's final state must satisfy (w.r.t. the PLD), and leave to the solver questions of how the language-acquisition device goes from the initial state to the final state and what that final state is.

Acknowledgements

I would like to thank three anonymous reviewers as well as Robert C. Berwick, Sandiway Fong and Sanjoy Mitter for their comments and feedback.

[5]As presented in (Adger, 2003; Hornstein et al., 2005; Radford, 2009; Collins and Stabler, 2016).

[6]Cf. Neural-network based UD parsing frameworks that have difficulty generalizing from degree-0/1 embedding sentences to correctly parse degree-n embedding sentences for $n \geq 2$. See (Indurkhya et al., 2021) for details.

References

David Adger. 2003. *Core syntax: A minimalist approach*, volume 33. Oxford University Press Oxford.

Robert C. Berwick. 1985. *The acquisition of syntactic knowledge*. MIT Press.

Robert C. Berwick, Paul Pietroski, Beracah Yankama, and Noam Chomsky. 2011. Poverty of the stimulus revisited. *Cognitive Science*, 35(7):1207–1242.

Noam Chomsky. 1965. *Aspects of the theory of syntax*. MIT Press.

Noam Chomsky. 1986. *Knowledge of language: Its nature, origin, and use*. Greenwood Publishing Group.

Noam Chomsky. 2013. Poverty of the stimulus: Willingness to be puzzled. In *Rich languages from poor inputs*, pages 61–67. Oxford University Press.

Chris Collins and Edward Stabler. 2016. A formalization of minimalist syntax. *Syntax*, 19(1):43–78.

Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient smt solver. In *Proceedings of the Theory and Practice of Software*, TACAS'08/ETAPS'08, pages 337–340, Berlin, Heidelberg. Springer-Verlag.

Leonardo De Moura and Nikolaj Bjørner. 2011. Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9):69–77.

Norbert Hornstein, Jairo Nunes, and Kleanthes K Grohmann. 2005. *Understanding minimalism*. Cambridge University Press.

Sagar Indurkhya. 2020. Inferring minimalist grammars with an SMT-solver. *Proceedings of the Society for Computation in Linguistics*, 3(1):476–479.

Sagar Indurkhya. 2021a. *Solving for syntax*. Ph.D. thesis, Massachusetts Institute of Technology.

Sagar Indurkhya. 2021b. Using collaborative filtering to model argument selection. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 629–639, Held Online. INCOMA Ltd.

Sagar Indurkhya, Beracah Yankama, and Robert C. Berwick. 2021. Evaluating Universal Dependency parser recovery of predicate argument structure via CompChain analysis. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 116–128, Online. Association for Computational Linguistics.

Andrew Radford. 2009. *An introduction to English sentence structure*. Cambridge University Press.

Manny Rayner, Asa Hugosson, and Goran Hagert. 1988. Using a logic grammar to learn a lexicon. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*.

Edward Stabler. 1996. Derivational minimalism. In *International Conference on Logical Aspects of Computational Linguistics*, pages 68–95. Springer.

Edward P Stabler. 2001. Recognizing head movement. In *International Conference on Logical Aspects of Computational Linguistics*, pages 245–260. Springer.