**ARE WE SMARTER THAN A THIRD GRADER?**

A Parameter-based Evolutionary Model for Child Language Acquisition

Scott Dubinsky, Emily Jones, Richard Higgins

How young children use native-language inputs to develop a functioning grammar is not clearly understood. We seek to provide an example model for this process, using the principles and parameters approach within Universal Grammar. When a child is born, language parameters are either unset or are set to default values. During a child’s critical period for language development, sentences they hear are parsed and used to set the parameters to the appropriate values for the language. Several models of this process have been proposed, but none satisfactorily solve the problem. We propose an evolutionary model for grammar development based on sentence input during the critical period, and will use Fodor’s theory of a supergrammar capable of parsing any sentence to provide us with input.

We hypothesize that if we evolve a grammar and vary different parental selection methods such as elitism, the recombination frequency and location, the mutation frequency and location, and survivor selection, then we will avoid issues alternate approaches have including large memory storage requirements, local maxima issues, an overdependence on default values, and the empirically unfounded tendency to predict rapid grammar change when a single sentence is heard.

In our genetic algorithm, which will be implemented in a scripting language, agents will be grammars, each grammar being a set of parameters. Our genotypes will be initially random bitstrings with parameters set to 0 or 1. This models the idea that children store parameters rather than sentences as they develop. During each round, a schema will be generated that sets some parameters to the value of the language’s grammar. All unset parameters will be represented by \*. This schema represents an input sentence which has already been lexed and parsed into its parameters. Different parameters will be set in the input schema based probabilistically to model how different parameters appear at different frequencies in a language. The number of parameters which can be set in a schema will be determined by an average parameter density value. The phenotype will be the ability of an agent’s grammar to match the input grammar. The fitness function will be the Hamming distance between the agent’s grammar and the input grammar. This model’s a child’s ability to match the parameters set in his or her grammar to those set by a sentence the child hears. After the fitness is calculated, the algorithm will use Roulette wheel selection to determine which grammars will be used in reproduction; these grammars will be recombined and mutated with set probabilities, then a new generation of agents will be produced using full replacement. The algorithm will terminate when the best-scoring grammar bit-string remains the same over a set number of generations, and success of an algorithm run can be measured by the best Hamming distance between an agent’s grammar and the language’s grammar.

Our grammars, input schemas, and probabilities will be set randomly, but can be explained best through a real-world example. For instance, consider a child’s grammar for the English language might be stored as “01001010100.” Given an English language grammar of “11\*010\*0101”, an input sentence “Bob runs” is generated. This would set the parameter “subject comes first,” which would overall be set in the input schema with probability 99% due to its frequent appearance in the language, but not the parameter “direct object comes after verb,” which might be set with probability 80%. Thus, the parsed and lexed sentence would produce the schema “11\*0\*\*\*010\*.” The child’s grammar, “01001010100,” has a Hamming distance of 1, as the grammar matches the input sentence on all but 1 parameter. Thus, this model of the grammar has a high probability of being used in the future, but might be paired with another grammar (recombination) or internally modified (mutation) before it is used again.

We will be modifying algorithm variables to determine which settings produce a grammar closest to the language’s grammar the fastest. As we determine the best values for each variable, we will extend the length of the grammar and decrease the parameter density to better model real language. Depending on time, we will also modify the algorithm to model various aspects of language acquisition. First, we could set some parameters incorrectly with a certain probability to model error, such as a child hearing an incorrect sentence. Second, we could decrease the number of agents to model memory limitations. Third, we could make certain parameters dependent on other parameters to model how some aspects of grammar are only seen observed other aspects are present. Next, we could allow each agent to have multiple grammars and give input strings from multiple languages, modeling children who are raised bilingually. Finally, we could examine the effects of negative feedback by only giving input schemas with errors, e.g. to model “I want bike,” and changing the fitness function to reward the grammars that match the input schema the least, e.g. to model the parent telling the child their sentence is wrong.