A quantitative evaluation of naturalistic models of language acquisition; the efficiency of the Triggering Learning Algorithm compared to a Categorial Grammar Learner

Paula Buttery
Natural Language and Information Processing Group,
Computer Laboratory, Cambridge University,
15 JJ Thomson Avenue, Cambridge, CB3 0FD, UK
paula.buttery@cl.cam.ac.uk

Abstract

Naturalistic theories of language acquisition assume learners to be endowed with some innate language knowledge. The purpose of this innate knowledge is to facilitate language acquisition by constraining a learner's hypothesis space. This paper discusses a naturalistic learning system (a Categorial Grammar Learner (CGL)) that differs from previous learners (such as the Triggering Learning Algorithm (TLA) (Gibson and Wexler, 1994)) by employing a dynamic definition of the hypothesis-space which is driven by the Bayesian Incremental Parameter Setting algorithm (Briscoe, 1999). We compare the efficiency of the TLA with the CGL when acquiring an independently and identically distributed English-like language in noiseless conditions. We show that when convergence to the target grammar occurs (which is not guaranteed), the expected number of steps to convergence for the TLA is shorter than that for the CGL initialized with uniform priors. However, the CGL converges more reliably than the TLA. We discuss the trade-off of efficiency against more reliable convergence to the target grammar.

1 Introduction

A normal child acquires the language of her environment without any specific training. Chomsky (1965) claims that, given the "relatively slight exposure" to examples and "remarkable complexity" of language, it would be "an extraordinary intellectual achievement" for a child to acquire a language if not specifically designed to do so. His *Argument from the Poverty of the Stimulus* suggests that if we know X, and X is undetermined by learning experience then X must be innate. For an example consider structure dependency in language syntax:

A question in English can be formed by inverting the auxiliary verb and subject noun-phrase: (1a) "*Dinah* **was** drinking a saucer of milk"; (1b) "**was** *Dinah* drinking a saucer of milk?"

Upon exposure to this example, a child could hypothesize infinitely many question-formation rules, such as: (i) *swap the first and second words in the sentence*; (ii) *front the first auxiliary verb*; (iii) *front words beginning with w*.

The first two of these rules are refuted if the child encounters the following: (2a) "*the cat who was grinning at Alice* **was** disappearing"; (2b) "**was** *the cat who was grinning at Alice* disappearing?"

If a child is to converge upon the correct hypothesis unaided she must be exposed to sufficient examples so that all false hypotheses are refuted. Unfortunately such examples are not readily available in child-directed speech; even the constructions in examples **(2a)** and **(2b)** are rare (Legate, 1999). To compensate for this lack of data Chomsky suggests that some principles of language are already available in the child's mind. For example, if the child had innately "known" that all grammar rules are structurally-dependent upon syntax she would never have hypothesized rules **(i)** and **(iii)**. Thus, Chomsky theorizes that a human mind contains a Universal Grammar which defines a hypothesis-space of "legal" grammars.[1] This hypothesis-space must be both large enough to contain grammar's for all of the world's languages and small enough to ensure successful acquisition given the sparsity of data. Language acquisition is the process of searching the hypothesis-space for the grammar that most closely describes the language of the environment. With estimates of the number of living languages being around 6800 (Ethnologue, 2004) it is not sensible to model the hypothesis-space of grammars explicitly, rather it must be modeled parametrically. Language acquisition is then the process of setting these parameters. Chomsky (1981) suggested that parameters should represent points of variation between languages, however the only requirement for parameters is that they define the current hypothesis-space.

[1] Discussion of structural dependence as evidence of the Argument from the Poverty of Stimulus is illustrative, the significance being that innate knowledge in any form will place constraints on the hypothesis-space

The properties of the parameters used by this learner (the CGL) are as follows: (1) Parameters are lexical; (2) Parameters are inheritance based; (3) Parameter setting is statistical.

1 - Lexical Parameters

The CGL employs parameter setting as a means to acquire a lexicon; differing from other parametric learners, (such as the Triggering Learning Algorithm (TLA) (Gibson and Wexler, 1994) and the Structural Triggers Learner (STL) (Fodor, 1998b), (Sakas and Fodor, 2001)) which acquire general syntactic information rather than the syntactic properties associated with individual words.[2]

In particular, a categorial grammar is acquired. The syntactic properties of a word are contained in its lexical entry in the form of a syntactic category. A word that may be used in multiple syntactic situations (or sub-categorization frames) will have multiple entries in the lexicon.

Syntactic categories are constructed from a finite set of primitive categories combined with two operators ($/$ and $\backslash$) and are defined by their members ability to combine with other constituents; thus constituents may be thought of as either functions or arguments.

The arguments of a functional constituent are shown to the right of the operators and the result to the left. The forward slash operator ($/$) indicates that the argument must appear to the right of the function and a backward slash ($\backslash$) indicates that it must appear on the left. Consider the following CFG structure which describes the properties of a transitive verb:

s	$\rightarrow$	**np vp**
vp	$\rightarrow$	**tv np**
tv	$\rightarrow$	gets, finds, ...

Assume that there is a set of primitive categories $\{\mathbf{s}, \mathbf{np}\}$. A $\mathbf{vp}$ must be in the category of functional constituents that takes a $\mathbf{np}$ from the left and returns an $\mathbf{s}$. This can be written $\mathbf{s}\backslash\mathbf{np}$. Likewise a $\mathbf{tv}$ takes an $\mathbf{np}$ from the right and returns a $\mathbf{vp}$ (whose type we already know). A $\mathbf{tv}$ may be written $(\mathbf{s}\backslash\mathbf{np})/\mathbf{np}$.

Rules may be used to combine categories. We assume that our learner is innately endowed with the rules of function application, function composition and generalized weak permutation (Briscoe, 1999) (see figures 1 and 2).

- Forward Application ($>$)
  $X/Y \ Y \rightarrow X$
- Backward Application ($<$)
  $Y \ X\backslash Y \rightarrow X$
- Forward Composition ($> B$)
  $X/Y \ Y/Z \rightarrow X/Z$
- Backward Composition ($< B$)
  $Y\backslash X \ Z\backslash Y \rightarrow X\backslash Z$
- Generalized Weak Permutation (P)
  $((X \mid Y_1)... \mid Y_n) \rightarrow ((X \mid Y_n)... \mid Y_1)$
  where $\mid$ is a variable over $\backslash$ and $/$.

[2]The concept of lexical parameters and the lexical-linking of parameters is to be attributed to Borer (1984).

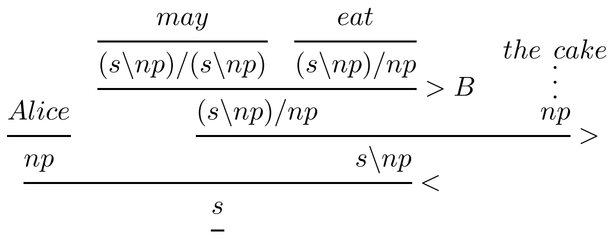

Figure 1: Illustration of forward/backward application ($>$, $<$) and forward composition ($> B$)

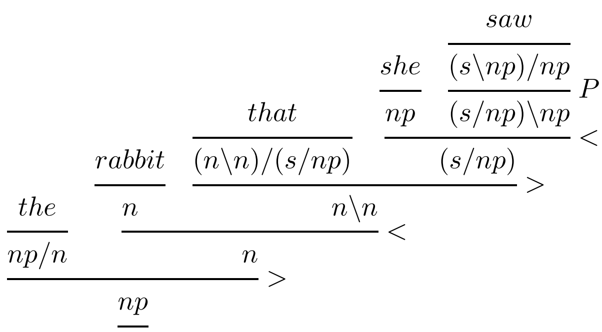

Figure 2: Illustration of generalized weak permutation (P)

The lexicon for a language will contain a finite subset of all possible syntactic categories, the size of which depends on the language. Steedman (2000) suggests that for English the lexical functional categories never need more than five arguments and that these are needed only in a limited number of cases such as for the verb *bet* in the sentence *I bet you five pounds for England to win.*

The categorial grammar parameters of the CGL are concerned with defining the set of syntactic categories present in the language of the environment. Converging on the correct set aids acquisition by constraining the learner's hypothesized syntactic categories for an unknown word. A parameter (with value of either ACTIVE or INACTIVE) is associated with every possible syntactic category to indicate whether the learner considers the category to be part of the target grammar.

Some previous parametric learners (TLA and STL) have been primarily concerned with overall syntactic phenomena rather than the syntactic properties of individual words. Movement parameters (such as the $V2$ parameter of the TLA) may be captured by the CGL using innate rules or multiple lexical entries. For instance, Dutch and German word order is captured by assuming that verbs in these languages systematically have two categories, one determining main clause order and the other subordinate clause orders.

2 - Inheritance Based Parameters

The complex syntactic categories of a categorial grammar are a sub-categorization of simpler categories; consequently categories may be arranged in a hierarchy with more complex categories inheriting from simpler ones. Figure 3 shows a fragment of a possible hierarchy. This hierarchical organization of parameters provides the learner with several benefits: (1) The hierarchy can enforce an order on learning; constraints may be imposed such that a parent parameter must be acquired before a child parameter (for example, in Figure 3, the learner must acquire intransitive verbs before transitive verbs may be hypothesized). (2) Parameter values may be inherited as a method of acquisition. (3) The parameters are stored efficiently.

- s - ACTIVE
  - s/s
  - $s\backslash np$ - ACTIVE
    - $[s\backslash np]/np$ - ACTIVE
    - $[s\backslash np]/[s\backslash np]$

Figure 3: Partial hierarchy of syntactic categories. Each category is associated with a parameter indicating either ACTIVE or INACTIVE status.

3 - Statistical Parameter Setting

The learner uses a statistical method to track relative frequencies of parameter-setting-utterances in the input.[3] We use the Bayesian Incremental Parameter Setting (BIPS) algorithm (Briscoe, 1999) to set the categorial parameters. Such an approach sets the parameters to the values that are most likely given all the accumulated evidence. This represents a compromise between two extremes: implementations of the TLA are memoryless allowing a parameter values to oscillate; some implementations of the STL set a parameter once, for all time.

Using the BIPS algorithm, evidence from an input utterance will either strengthen the current parameter settings or weaken them. Either way, there is re-estimation of the probabilities associated with possible parameter values. Values are only assigned when sufficient evidence has been accumulated, i.e. once the associated probability reaches a threshold value. By employing this method, it becomes unlikely for parameters to switch between settings as the consequence of an erroneous utterance.

Another advantage of using a Bayesian approach is that we may set default parameter values by assigning Bayesian priors; if a parameter's default value is strongly biased against the accumulated evidence then it will be difficult to switch. Also, we no longer need to worry about ambiguity in parameter-setting-utterances (Clark, 1992) (Fodor, 1998b): the Bayesian approach allows us to solve this problem "for free" since indeterminacy just becomes another case of error due to misclassification of input data (Buttery and Briscoe, 2004).

2 Overview of the Categorial Grammar Learner

The learning system is composed of a three modules: a semantics learning module, syntax learning module and memory module. For each utterance heard the learner receives an input stream of word tokens paired with possible semantic hypotheses. For example, on hearing the utterance "Dinah drinks milk" the learner may receive the pairing: ({*dinah*, *drinks*, *milk*}, **drinks(dinah, milk)**).

2.1 The Semantic Module

The semantic module attempts to learn the mapping between word tokens and semantic symbols, building a lexicon containing the meaning associated with each word sense. This is achieved by analyzing each input utterance and its associated semantic hypotheses using cross-situational techniques (following Siskind (1996)).

For a trivial example consider the utterances "Alice laughs" and "Alice eats cookies"; they might have word tokens paired with semantic expressions as follows: ({*alice*, *laughs*}, **laugh(alice)**), ({*alice*, *eats*, *cookies*}, **eat(alice, cookies)**).

From these two utterances it is possible to ascertain that the meaning associated with the word token *alice* must be **alice** since it is the only semantic element that is common to both utterances.

[3] Other statistical parameter setting models include Yang's Variational model (2002) and the Guessing STL (Fodor, 1998a)

2.2 The Syntactic Module

The learning system links the semantic module and syntactic module by using a typing assumption: *the semantic arity of a word is usually the same as its number of syntactic arguments.* For example, if it is known that *likes* maps to $\mathbf{like}(\mathbf{x}, \mathbf{y})$, then the typing assumption suggests that its syntactic category will be in one of the following forms: $a\backslash b\backslash c$, $a/b\backslash c$, $a\backslash b/c$, $a/b/c$ or more concisely $a \mid b \mid c$ (where a, b and c may be basic or complex syntactic categories themselves).

By employing the typing assumption the number of arguments in a word's syntactic category can be hypothesized. Thus, the objective of the syntactic module is to discover the arguments' category types and locations.

The module attempts to create valid parse trees starting from the syntactic information already assumed by the typing assumption (following Buttery (2003)). A valid parse is one that adheres to the rules of the categorial grammar as well as the constraints imposed by the current settings of the parameters. If a valid parse can not be found the learner assumes the typing assumption to have failed and backtracks to allow type raising.

2.3 Memory Module

The memory module records the current state of the hypothesis-space. The syntactic module refers to this information to place constraints upon which syntactic categories may be hypothesized. The module consists of two hierarchies of parameters which may be set using the BIPS algorithm:

Categorial Parameters determine whether a category is in use within the learner's current model of the input language. An inheritance hierarchy of all possible syntactic categories (for up to five arguments) is defined and a parameter associated with each one (Villavicencio, 2002). Every parameter (except those associated with primitive categories such as S) is originally set to INACTIVE, i.e. no categories (except primitives) are known upon the commencement of learning. A categorial parameter may only be set to ACTIVE if its parent category is already active and there has been satisfactory evidence that the associated category is present in the language of the environment.

Word Order Parameters determine the underlying order in which constituents occur. They may be set to either FORWARD or BACKWARD depending on whether the constituents involved are generally located to the right or left. An example is the parameter that specifies the direction of the subject of a verb: if the language of the environment is English this parameter would be set to BACKWARD since subjects generally appear to the left of the verb. Evidence for the setting of word order parameters is collected from word order statistics of the input language.

3 The acquisition of an English-type language

The English-like language of the three-parameter system studied by Gibson and Wexler has the parameter settings and associated unembedded surface-strings as shown in Figure 4. For this task we assume that the surface-strings of the English-like language are independent and identically distributed in the input to the learner.

Specifier	Complement	V2
0 (*Left*)	1 (*Right*)	0 (*off*)

1. Subj Verb
2. Subj Verb Obj
3. Subj Verb Obj Obj
4. Subj Aux Verb
5. Subj Aux Verb Obj
6. Subj Aux Verb Obj Obj
7. Adv Subj Verb
8. Adv Subj Verb Obj
9. Adv Subj Verb Obj Obj
10. Adv Subj Aux Verb
11. Adv Subj Aux Verb Obj
12. Adv Subj Aux Verb Obj Obj

Figure 4: Parameter settings and surface-strings of Gibson and Wexler's English-like Language.

3.1 Efficiency of Trigger Learning Algorithm

For the TLA to be successful it must converge to the correct parameter settings of the English-like language. Berwick and Niyogi (1996) modeled the TLA as a Markov process (see Figure 5).

Using this model it is possible to calculate the probability of converging to the target from each starting grammar and the expected number of steps before convergence.

Probability of Convergence:

Consider starting from Grammar 3, after the process finishes looping it has a $3/5$ probability of moving to Grammar 4 (from which it will never converge) and a $2/5$ probability of moving to Grammar 7 (from which it will definitely converge), therefore there is a 40% probability of converging to the target grammar when starting at Grammar 3.

Expected number of Steps to Convergence:
Let S_n be the expected number of steps from state n to the target state. For starting grammars 6, 7 and 8, which definitely converge, we know:

$$S_6 = 1 + \frac{5}{6} S_6 \quad (1)$$

$$S_7 = 1 + \frac{2}{3} S_7 + \frac{1}{18} S_8 \quad (2)$$

$$S_8 = 1 + \frac{1}{12} S_6 + \frac{1}{36} S_7 + \frac{8}{9} S_8 \quad (3)$$

and for the times when we do converge from grammars 3 and 1 we can expect:

$$S_1 = 1 + \frac{3}{5} S_1 \quad (4)$$

$$S_3 = 1 + \frac{31}{33} S_3 \quad (5)$$

Figure 6 shows the probability of convergence and expected number of steps to convergence for each of the starting grammars. The expected number of steps to convergence ranges from infinity (for starting grammars 2 and 4) down to 2.5 for Grammar 1. If the distribution over the starting grammars is uniform then the overall probability of converging is the sum of the probabilities of converging from each state divided by the total number of states:

$$\frac{1.00 + 1.00 + 1.00 + 1.00 + 0.40 + 0.66}{8} = 0.63 \quad (6)$$

and the expected number of steps given that you converge is the weighted average of the number of steps from each possibly converging state:

$$\frac{5.47 + 14.87 + 6 + 21.98 \times 0.4 + 2.5 \times 0.66}{1.00 + 1.00 + 1.00 + 1.00 + 0.40 + 0.66} = 7.26 \quad (7)$$

3.2 Efficiency of Categorial Grammar Learner

The input data to the CGL would usually be an utterance annotated with a logical form; the only data available here however, is surface-strings consisting of word types. Hence, for the purpose of comparison with the TLA the semantic module of our learner is by-passed; we assume that mappings to semantic forms have previously been acquired and that the subject and objects of surface-strings are known. For example, given surface-string 1 (*Subj Verb*) we assume the mapping *Verb* $\mapsto$ **verb(x)**, which provides *Verb* with a syntactic category of the form $a|b$ by the typing assumption (where a, b are unknown syntactic categories and $|$ is an operator over $\backslash$ and $/$); we also assume *Subj* to map to a primitive syntactic category SB, since it is the subject of *Verb*.

The criteria for success for the CGL when acquiring Gibson and Wexler's English-like language is a lexicon containing the following:[4]

Adv	S/S	**Aux**	$[S\backslash SB]/[S\backslash SB]$
Obj	OB	**Verb**	$S\backslash SB$
Subj	SB		$[S\backslash SB]/OB$
			$[[S\backslash SB]/OB]/OB$

where S (sentence), SB (subject) and OB (object) are primitive categories which are innate to the learner with SB and OB assumed to be derivable from the semantic module.

During the learning process the CGL will have constructed a category hierarchy by setting appropriate categorial parameters to true (see Figure 7). The learner will have also constructed a word-order hierarchy (Figure 8), setting parameters to FORWARD or BACKWARD. These hierarchies are used during the learning process to constrain hypothesized syntactic categories. For this task the setting of the word-order parameters becomes trivial and their role in constraining hypotheses negligible; consequently, the rest of our argument will relate to categorial parameters only. For the purpose of this

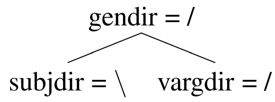

Figure 8: Word-order parameter settings required to parse Gibson and Wexler's English-like language.

analysis parameters are initialized with uniform priors and are originally set INACTIVE. Since the input is noiseless, the switching threshold is set such that parameters may be set ACTIVE upon the evidence from one surface-string.

It is a requirement of the parameter setting device that the parent-types of hypothesized syntax categories are ACTIVE before new parameters are set. Thus, the learner is not allowed to hypothesize the syntactic category for a transitive verb $[[S\backslash SB]/OB]$ before it has learnt the category for an intransitive verb $[S\backslash SB]$; this behaviour constrains over-generation. Additionally, it is usually not possible to derive a word's full syntactic category (i.e. without any remaining unknowns) unless it is the only new word in the clause.

As a consequence of these issues, the order in which the surface-strings appear to the learner af-

[4]Note that the lexicon would usually contain orthographic entries for the words in the language rather than word type entries.

fects the speed of acquisition. For instance, the learner prefers to see the surface-string *Subj Verb* before *Subj Verb Obj* so that it can acquire the maximum information without wasting any strings. For the English-type language described by Gibson and Wexler the learner can optimally acquire the whole lexicon after seeing only 5 surface-strings (one string needed for each new complex syntactic category to be learnt). However, the strings appear to the learner in a random order so it is necessary to calculate the expected number of strings (or steps) before convergence.

The learner must necessarily see the string *Subj Verb* before it can learn any other information. With 12 surface-strings the probability of seeing *Subj Verb* is $1/12$ and the expected number of strings before it is seen is 12. The learner can now learn from 3 surface-strings: *Subj Verb Obj*, *Subj Aux Verb* and *Adv Subj Verb*. Figure 9 shows a Markov structure of the process. From the model we can calculate the expected number of steps to converge to be 24.53.

4 Conclusions

The TLA and CGL were compared for efficiency (expected number of steps to convergence) when acquiring the English-type grammar of the three-parameter system studied by Gibson and Wexler. The expected number of steps for the TLA was found to be 7.26 but the algorithm only converged 63% of the time. The expected number of steps for the CGL is 24.53 but the learner converges more reliably; a trade off between efficiency and success. With noiseless input the CGL can only fail if there is insufficient input strings or if Bayesian priors are heavily biased against the target. Furthermore, the CGL can be made robust to noise by increasing the probability threshold at which a parameter may be set ACTIVE; the TLA has no mechanism for coping with noisy data.

The CGL learns incrementally; the hypothesis-space from which it can select possible syntactic categories expands dynamically and, as a consequence of the hierarchical structure of parameters, the speed of acquisition increases over time. For instance, in the starting state there is only a $1/12$ probability of learning from surface-strings whereas in state k (when all but one category has been acquired) there is a $1/2$ probability. It is likely that with a more complex learning task the benefits of this incremental approach will outweigh the slow starting costs. Related work on the effects of incremental learning on STL performance (Sakas, 2000) draws similar conclusions. Future work hopes to compare the CGL with other parametric learners (such as the STL) in larger domains.

References

R Berwick and P Niyogi. 1996. Learning from triggers. *Linguistic Inquiry*, 27(4):605–622.

H Borer. 1984. *Parametric Syntax: Case Studies in Semitic and Romance Languages*. Foris, Dordrecht.

E Briscoe. 1999. The acquisition of grammar in an evolving population of language agents. *Machine Intelligence*, 16.

P Buttery and T Briscoe. 2004. The significance of errors to parametric models of language acquisition. Technical Report SS-04-05, American Association of Artificial Intelligence, March.

P Buttery. 2003. A computational model for first language acquisition. In *CLUK-6*, Edinburgh.

N Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press.

N Chomsky. 1981. *Lectures on Government and Binding*. Foris Publications.

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

Ethnologue. 2004. Languages of the world, 14th edition. SIL International. http://www.ethnologue.com/.

J Fodor. 1998a. Parsing to learn. *Journal of Psycholinguistic Research*, 27(3):339–374.

J Fodor. 1998b. Unambiguous triggers. *Linguistic Inquiry*, 29(1):1–36.

E Gibson and K Wexler. 1994. Triggers. *Linguistic Inquiry*, 25(3):407–454.

J Legate. 1999. Was the argument that was made empirical? Ms, Massachusetts Institute of Technology.

W Sakas and J Fodor. 2001. The structural triggers learner. In S Bertolo, editor, *Language Acquisition and Learnability*, chapter 5. Cambridge University Press, Cambridge, UK.

W Sakas. 2000. *Ambiguity and the Computational Feasibility of Syntax Acquisition*. Ph.D. thesis, City University of New York.

J Siskind. 1996. A computational study of cross situational techniques for learning word-to-meaning mappings. *Cognition*, 61(1-2):39–91, Nov/Oct.

M Steedman. 2000. *The Syntactic Process*. MIT Press/Bradford Books.

A Villavicencio. 2002. *The acquisition of a unification-based generalised categorial grammar*. Ph.D. thesis, University of Cambridge.

C Yang. 2002. *Knowledge and Learning in Natural Language*. Oxford University Press.

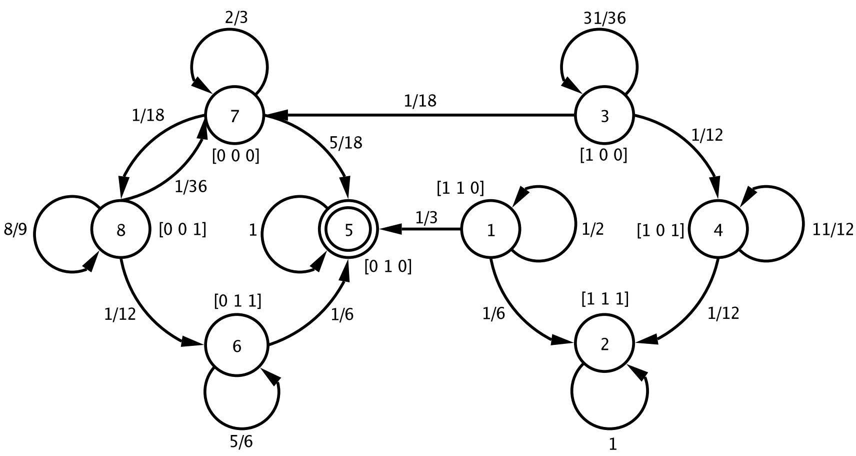

Figure 5: Gibson and Wexler's TLA as a Markov structure. Circles represent possible grammars (a configuration of parameter settings). The target grammar lies at the centre of the structure. Arrows represent the possible transitions between grammars. Note that the TLA is constrained to only allow movement between grammars that differ by one parameter value. The probability of moving between Grammar G_i and Grammar G_j is a measure of the number of target surface-strings that are in G_j but not G_i normalized by the total number of target surface-strings as well as the number of alternate grammars the learner can move to. For example the probability of moving from Grammar 3 to Grammar 7 is $2/12 * 1/3 = 1/18$ since there are 2 target surface-strings allowed by Grammar 7 that are not allowed by Grammar 3 out of a possible of 12 and three grammars that differ from Grammar 3 by one parameter value.

Initial Language	Initial Grammar	Prob. of Converging	Expected no. of Steps
VOS -V2	110	0.66	2.50
VOS +V2	111	0.00	n/a
OVS -V2	100	0.40	21.98
OVS +V2	101	0.00	n/a
SVO -V2	010	1.00	0.00
SVO +V2	011	1.00	6.00
SOV -V2	000	1.00	5.47
SOV +V2	001	1.00	14.87

Figure 6: Probability and expected number of steps to convergence from each starting grammar to an English-like grammar (SVO -V2) when using the TLA.

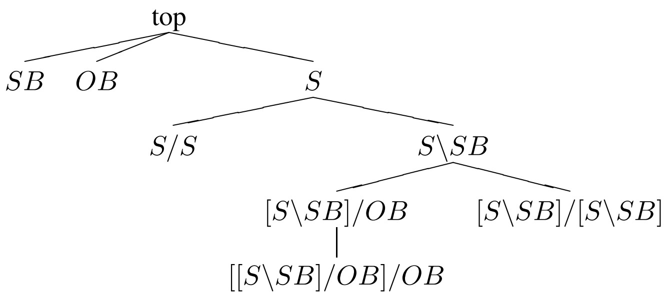

Figure 7: Category hierarchy required to parse Gibson and Wexler's English-like language.

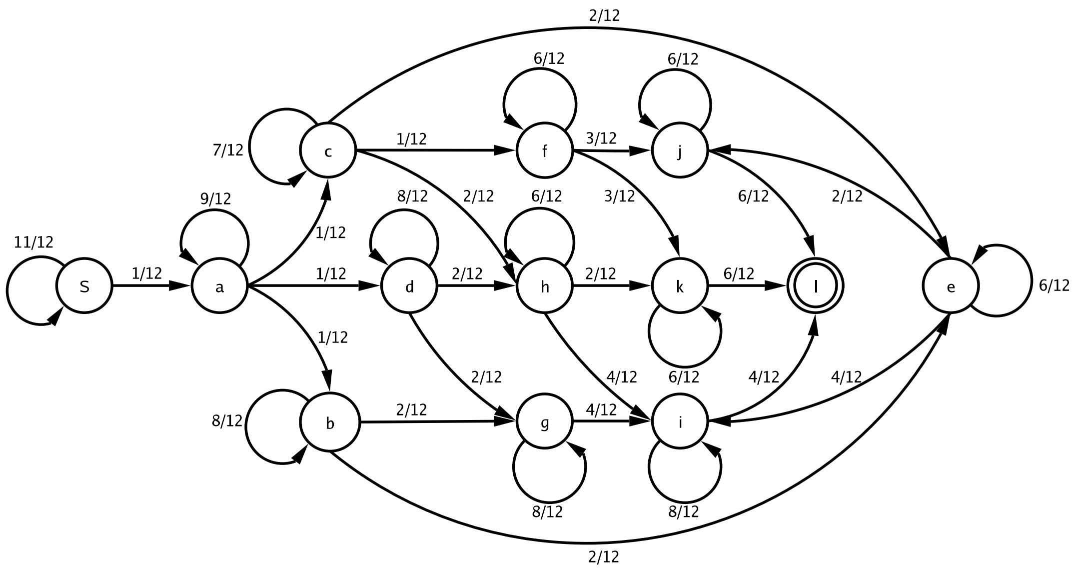

Figure 9: The CGL as a Markov structure. The states represent the set of known syntactic categories: state S - $\{\}$, state a - $\{S\backslash SB\}$, state b - $\{S\backslash SB,\ S/S\}$, state c - $\{S\backslash SB,\ [S\backslash SB]/OB\}$, state d - $\{S\backslash SB,\ [S\backslash SB]/[S\backslash SB]\}$, state e - $\{S\backslash SB,\ S/S,\ [S\backslash SB]/OB\}$, state f - $\{S\backslash SB,\ [S\backslash SB]/OB,\ [[S\backslash SB]/OB]/OB\}$, state g - $\{S\backslash SB,\ [S\backslash SB]/[S\backslash SB],\ S/S\}$ state h - $\{S\backslash SB,\ [S\backslash SB]/[S\backslash SB],\ [S\backslash SB]/OB\}$, state i - $\{S\backslash SB,\ S/S,\ [S\backslash SB]/OB,\ [S\backslash SB]/[S\backslash SB]\}$, state j - $\{S\backslash SB,\ S/S,\ [S\backslash SB]/OB,\ [[S\backslash SB]/OB]/OB\}$, state k - $\{S\backslash SB,\ [S\backslash SB]/OB,\ [[S\backslash SB]/OB]/OB,\ [S\backslash SB]/[S\backslash SB]\}$, state l - $\{S\backslash SB,\ [S\backslash SB]/OB,\ [[S\backslash SB]/OB]/OB,\ [S\backslash SB]/[S\backslash SB],\ S/S\}$.