Categorial Module Grammar Learning in Gold's Model (Summary)

Student: Camilo Thorne

Instructor: Dénis Béchet
Assistant Professor – Computer Science.
University of Paris 13, Villetaneuse, France.
University of Nantes, Nantes, France.

The aim of this master thesis was twofold. On the one hand, to study the formal properties of categorial module grammars. On the other, to see wether the class of grammars thus defined were learnable through identification on the limit from positive examples (Gold's learning model). The work was carried at the LIPN (Laboratoire d'Informatique de Paris Nord) under the direction of Mr. D. Béchet.

The first sections of this master thesis were devoted to categorial grammar and linear logic. Categorial module grammars (GC_{mod} and GC_{modf}) are a class of categorial grammars (and therefore of lexicalised grammars) in which instead of the Lambek Calculus (LC) as type system, one works within the framework of Linear Logic (LL) proof-nets. In this latter formalism, to decide if a sequent is a theorem of the system (i.e. of LL) we build a proof-net, a graph from which a sequential proof tree can be extracted. The advantange being that cut-elimination is confluent for proof-nets (uniqueness of normal form). Categorial grammars are triples of the form $G = \langle \Sigma, \delta, S \rangle$, where Σ is an alphabet (a set of lexemes), S a distinguished type (that of well-formed chains) and δ a typing function. The grammars thus work by typing the words belonging to Σ and by looking for a proof for the sequent $A_1, ..., A_n \vdash S$, given a chain of n words $s = m_1...m_n$ typed by $A_1, ..., A_n$ – i.e. to know if $s = m_1...m_n \in L(G)$ we just see if the sequent $\delta(m_1), ..., \delta(m_n) \vdash S$ has a proof. The parse

of the chain (hence of a sentence or phrase) is then either a derivation tree for the associated types (the words $m_1, ..., m_n$ can be, for instance, seen as its yield) or a proof-net. Types code the syntactic category of words (nouns, transitive verbs, articles, pronouns, etc.). Categorial grammar is an alternative formal model for natural language syntax, i.e. w.r.t. the Chomsky model. This model was first put forth (amid others) by Lambek in [8]. I thus studied in detail how LC proves to be a subsystem of LL – it is in fact equivalent to Non-Commutative Multiplicative Intuitionistic Linear Logic ($MINCLL$). Henceforth, proof-nets were adapted to this new calculus, following Retoré (cf. [10], [9], [11] and [12]), who established that the class of planar intuitionistic proof-nets coincides exactly with that of $MINCLL$ proof-trees or derivations. As they are proof-nets, sequentialisation still applies to them. The decision procedure thus consists in building the sequent's proof structure (a graph built from the syntactic forest of the sequent's types and axiom links), looking for its correction graphs and veryfing their satisfying the so called correcteness criteria (acyclicity and connectness). This analysis gives way in its turn to modules, which are but partial proof-nets (i.e. partial subgraphs of the proof-net devoid of internal components).

It is on the last sections that categorial module grammars were defined and learning results proved. Categorial module grammars are grammars defined over something like structure languages: they are defined on module languages – i.e. whose lexemes have been annotated by modules (this is important: learning in Gold's model is impossible from bare strings or chains). This gives place to two classes: to grammars having an infinite alphabet (GC_{mod}) and grammars having a finite alphabet (GC_{modf}). Grammars are still finitary objets since the domain of δ is always finite and that δ determines them completely. Their new alphabet is $\Sigma \times \mathcal{M}$, where $\mathcal{M}$ is the set of modules, for any set Σ. Intuitively, this means that we have more than enough data as to perform identification in the limit from positive examples. This latter issue was pursued through some well-known sufficient conditions: finite elasticity and bounded finite thickness (which implies finite elasticity). In general, what we need is: (i) A grammar system $GS = \langle S^*, GK, L \rangle$ where S^* is the space of positive samples, GK a r.e. class of grammars and L the language function; and: (ii) A learning algorithm ϕ_{GK} which infers from a finite number of positive samples (an enumeration) of some $L \in \mathcal{L}(GK)$ (the class of languages generated by GK's grammars) some grammar $G_{\phi_{GK}}$, which converges to $G_{\phi_{GK}}$ thereafter. However, results by Angluin, Kanazawa

(finite elasticity, cf. [6]) and Shinohara (bounded finite thickness, cf. [14] and [13]) permit us to reason on structural properties of the language classes, the grammar classes and the grammar systems. I arrived to the following learning results:

- That the class $\mathcal{L}(GC_{modf})$, i.e. of languages generated by categorial module grammars of finite alphabet, has finite elasticity and hence GG_{modf} is learnable in Gold's model.
- That the grammar system $GS_{mod} = \langle(\Sigma \times \mathcal{M})^+, GC_{mod}, L\rangle$, i.e. that of categorial module grammars of infinite alphabet, has bounded finite thickness and that hence, for any $n \in \mathbb{N}$, the class $\{G \in GC_{mod} | t(G) \leq n,$ G reduced w.r.t. $\mathcal{X} \subseteq (\Sigma \times \mathcal{M})^+\}$ is learnable in Gold's model.
- That the class $\mathcal{L}(GC_{mod})$ has infinite elasticity and that therefore to show its learnability a learning algorithm should be constructed. This wasn't done due to lack of time.

Finally, I devoted some lines to classical categorial grammars (a.k.a. AB-grammars) and implemented them in SCHEME (a dialect of Lisp). As this wasn't the main objective of the work, I did it in an appendix.

As a matter of course, many problems were left open, both from a mathematical and a linguistical point of view – for the latter wasn't worked in depth. For instance, to study the semantics associated to proof-nets (in fact, to those equivalent to LC derivations). Girard has proposed a denotational semantics: the coherent space semantics (coherent spaces are a kind of category, a refinement of cartesian closed categories for intuitionistic logic). As well as an algebraical one, the click semantics. This point is particularly interesting, for it amounts to a concurrent model to Tarskian ("extensional") semantics that some logicians have advanced as the main basis for a natural language theory of meaning. As well as for as for Montague ("intensional") semantics, the other paradigm. On the other hand, the modules studied here are supposed to capture syntactical information and not semantical information – i.e. learnability (or unlearnability) results should have been extended to commutative types. Another important issue would have been to see up to what point this model (if appropiate for natural language) can shed some light on the links and relations existing between natural language syntax and

semantics.

Keywords: Categorial grammar, computational linguistics, learning in Gold's model, formal syntax, linear logic.

References

[1] Denis Bechet. Incremental parsing of lambek calculus using proof-net interphases. http://www-lipn.univ-paris13.fr/∼bechet, 2002.

[2] Denis Bechet et Annie Foret. k-valued non-associative lambek grammars are learnable from function-argument structures. *Electronic Notes in Theoretical Computer Science*, (84), 2003.

[3] Jacques Chazarain. *Programmer avec SCHEME. De la pratique à la théorie*. International Thomson Publishing, 1996.

[4] Antoine Cornuejols et Yves Kodratoff. *Apprentissage artificiel. Concepts et algorithmes*. Eyrolles, 2002.

[5] John E. Hopcroft et Jeffrey D. Ullman. *An Introduction to Automata Theory, Languages and Computation*. Addison Welesley, 1979.

[6] Makoto Kanazawa. *Learnable Classes of Categorial Grammars*. CSLI, 1998.

[7] Makoto Kanazawa. Lambek calculus: Recognizing power and complexity. http://www.illa.uva.nl/j50/contribs/kanazawa/∼1999, 1999.

[8] Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958.

[9] Christian Retore. Calcul de lambek et logique linéaire. *T.A.L.*, 37(2):39–70.

[10] Christian Retore. Logique linéaire et syntaxe des langues. Technical report, Université de Nantes et INRIA, 2002. (Habilitation à diriger des recherches).

[11] Christian Retore et André Lecomte. Words as modules and modules as partial proof-nets in a lexicalised grammar. In Carlos Martin-Vide, editor, *Mathematical and Computational Analysis of Natural Language*, volume 45 of *Functional and Structural Linguistics*. John Benjamins, 1998.

[12] Christian Retore et François Lemarche. Proof nets for the lambek calculus – an overview. In *Proceedings of the 1996 Roma Workshop 'Proofs and Linguistic Categories – Applications of Logic to the Analysis and Implemenation of Natural Language'*, 1996.

[13] Takeshi Shinohara. Inductive inference from positive data is powerful. In *Annual Workshop on Computational Learning Theory*, volume 3, 1990.

[14] Takeshi Shinohara. Inductive inference of monotonic formal systems from positive data. In *International Workshop on Algorithmic Learning Theory*, number 1, 1990.

[15] Jacques Stern. *Fondements mathématiques de l'informatique*. Ediscience International, 1994.

[16] Hans-Joerg Tiede. *Deductive Systems and Grammars: Proofs as Grammatical Structures*. PhD thesis, Indiana University (Bloomington), 1999.

[17] Hans-Joerg Tiede. Proof theory and formal grammars. applications of normalization. In Benedikt Loewe Thoralf Rarsch, Wolfgang Malzkorn, editor, *Foundations of The Formal Sciences II: Applications of Mathematical Logic in Philosophy and Linguistics*. Kluwer, 2000.

[18] A. S. Troelstra et H. Schwichtenberg. *Basic Proof Theory*. Cambridge U. Press, 2000.