Automata and AB-Categorial Grammars

Isabelle Tellier

GRAppA & Inria Futurs, Lille (MOSTRARE project)
Université Charles de Gaulle- Lille 3, 59653 Villeneuve d'Ascq, France
isabelle.tellier@univ-lille3.fr

1 Introduction

AB-categorial grammars (CGs in the following) is a lexicalized formalism having the expressive power of ϵ-free context-free languages [1]. It has a long common history with natural language [2]. Here, we first relate unidirectional CGs to a special case of recursive transition networks [4]. We then illustrate how the structures produced by a CG can be generated by a *pair of recursive automata*.

2 Automata for Unidirectional Categorial Grammars

Definition 1. *Let $\mathcal{B}$ be a set of basic categories among which is the axiom $S \in \mathcal{B}$. $Cat(\mathcal{B})$ is the smallest set including $\mathcal{B}$ and every A/B and $B\backslash A$, for any A, B in $Cat(\mathcal{B})$. A CG $G \subset \Sigma \times Cat(\mathcal{B})$ is a finite relation between a vocabulary Σ and $Cat(\mathcal{B})$. In CGs, the syntactic rules are reduces to two rewriting schemes: FA (Forward Application): $A/B \quad B \rightarrow A$ and BA (Backward Application): $B \quad B\backslash A \rightarrow A$. The language generated by a CG is the set of strings in Σ^* corresponding to a string in $(Cat(\mathcal{B}))^*$ which reduces to S. Unidirectional CGs make an exclusive use of / (or of \). They can produce every ϵ-free CF language.*

Example 1. The classical unidirectional CGs recognizing $a^n b^n, n \geq 1$ are: $G_{FA} = \{\langle a, S/B\rangle, \langle a, (S/B)/S\rangle, \langle b, B\rangle\}$ and $G_{BA} = \{\langle a, A\rangle, \langle b, A\backslash S\rangle, \langle b, S\backslash(A\backslash S)\rangle\}$. They can respectively be represented by the "recursive automata" given in Figure 1.

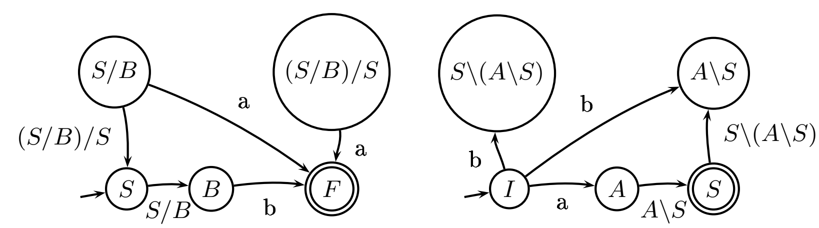

Fig. 1. Two recusive automata both recognizing $a^n b^n, n \geq 1$

In these automata (seee [3] for details), the transitions labelled by a state refer to *state languages*: for unidirectional CGs making only use of / (resp. of \), the language $L_{FA}(Q)$ (resp. $L_{BA}(Q)$) is the set of strings produced by starting in Q and reaching the state F (resp. by starting in I and reaching the state Q).

3 Automata for AB-Categorial Grammars

Now, to produce the same *structures* as a CG, it is enough to consider two mutually recursive automata: one for FA rules, the other for BA rules. $\forall Q \in Cat(\mathcal{B})$: $L(Q) = L_{FA}(Q) \cup L_{BA}(Q)$. This generative model improves the readability of a CG. A promising application domain is grammatical inference [3].

Example 2. Let $\mathcal{B} = \{S, T, CN\}$ (where T stands for "term" and CN for "common noun"), $\Sigma = \{John, runs, loves, a, cat\}$ and $G = \{\langle John, T\rangle, \langle loves, (T\backslash S)/T\rangle, \langle loves, T\backslash(S/T)\rangle, \langle runs, T\backslash S\rangle, \langle cat, CN\rangle, \langle a, (S/(T\backslash S))/CN\rangle, \langle a, ((S/T)\backslash S)/CN\rangle\}$.

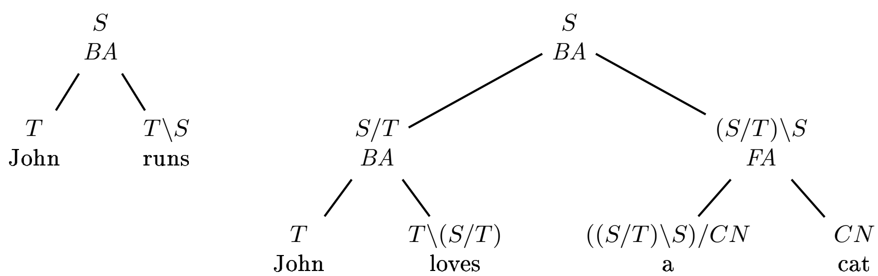

Fig. 2. Syntactic Parse Trees Produced by the Categorial Grammar G

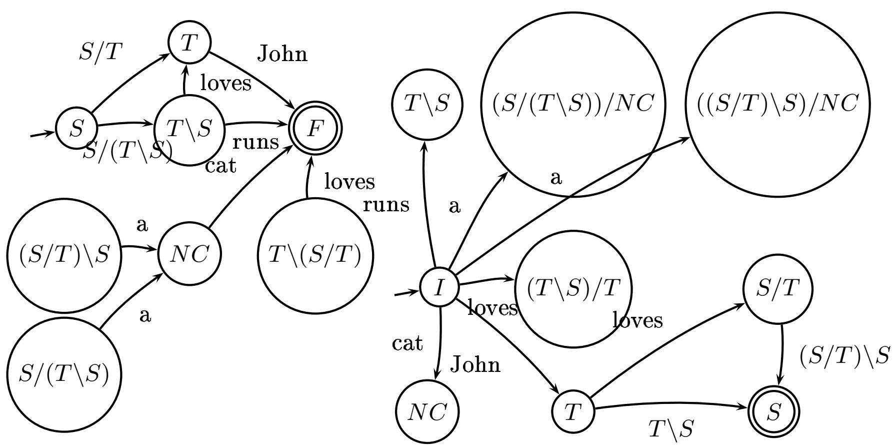

Fig. 3. A Pair of Mutually Recursive Automata Representing G

References

1. Y. Bar Hillel and C. Gaifman and E. Shamir. On Categorial and Phrase Structure Grammars. *Bulletin of the Research Council of Israel*, 9F, 1960.
2. R. T. Oehrle and E. Bach and D. Wheeler. *Categorial Grammars and Natural Language Structures* D. Reidel Publishing Company, Dordrecht, 1988.
3. I. Tellier *When Categorial Grammars meet Regular Grammatical Inference* proceedings of *LACL 2005*, LNAI 3492, p317-332, 2005.
4. W. A. Woods Transition network grammars for natural language analysis Communication of the ACM, vol.13, n.10, p591-606, 1970.