

Inférence grammaticale et grammaires catégorielles : vers la Grande Unification !

Isabelle Tellier

GRAppA & Inria Futurs, Lille
MOSTRARE project
Université Lille 3
59653 Villeneuve d'Ascq
France

`isabelle.tellier@univ-lille3.fr`

Résumé : Dans cet article, nous proposons de comparer les techniques employées en inférence grammaticale de langages réguliers par exemples positifs avec celles employées pour l'inférence de grammaires catégorielles. Pour cela, nous commençons par étudier la traduction entre automates et grammaires catégorielles, et inversement. Nous montrons ensuite que l'opérateur de généralisation utilisé pour l'apprentissage de grammaires catégorielles est strictement plus puissant que celui de fusion d'états (usuel en inférence grammaticale régulière), puisqu'il fait parfois sortir de la classe des langages réguliers. Nous proposons un nouveau modèle génératif qui généralise les automates à états finis pour représenter le résultat de cet opérateur. Nous montrons que ce modèle a au moins la même expressivité que les grammaires catégorielles unidirectionnelles, c'est-à-dire celle des langages algébriques. Enfin, nous exhibons une sous-classe des grammaires catégorielles unidirectionnelles pour laquelle l'apprentissage à partir de textes n'est presque pas plus coûteux que l'apprentissage à partir de structures.

1 Introduction

En inférence grammaticale, on étudie comment apprendre une grammaire à partir d'exemples de phrases qu'elle engendre -et de phrases qu'elle ne reconnaît pas, si des exemples négatifs sont disponibles. Les résultats fondateurs du domaine concernent l'apprenabilité de sous-classes de grammaires régulières, généralement représentées par des automates à états finis (Angluin, 1982; Oncina & Garcia, 1992; Dupont *et al.*, 1994; Denis *et al.*, 2002).

L'inférence de grammaires algébriques est évidemment plus difficile. Elle a fait l'objet de nombreux travaux, mais généralement plutôt empiriques. Or, nous aimerions l'envisager dans le modèle d'apprentissage à la limite par exemples positifs de Gold (Gold, 1967). Les résultats théoriques qui nous semblent les plus avancés du domaine sont dus à Kanazawa (Kanazawa, 1998), qui a prouvé dans ce modèle l'apprenabilité de sous-classes de grammaires catégorielles de type AB (un formalisme syntaxique qui a

la même expressivité que les grammaires algébriques). Ces résultats ont malheureusement peu de portée pratique, notamment parce que, à part dans des cas très particuliers, les algorithmes d'apprentissage auxquels ils donnent lieu sont très coûteux (Florêncio, 2001; Florêncio, 2002). Il semble que personne, jusqu'à présent, n'ait essayé de représenter par des grammaires catégorielles les langages réguliers, pour voir en quoi les résultats les concernant recourent ou non ceux connus en inférence de langages réguliers. C'est le point de départ de notre travail.

La première partie de cet article, après avoir introduit les définitions nécessaires, expose donc en détail comment passer d'un automate à états finis à une grammaire catégorielle, et réciproquement. Dans un deuxième temps, on tente d'utiliser cette traduction pour déduire des résultats d'apprenabilité nouveaux ou rapprocher des résultats obtenus indépendamment dans chacun des deux contextes. Mais la voie qui se révèle la plus prometteuse consiste à comparer ce que réalisent les algorithmes d'apprentissage eux-mêmes. Nous montrons ainsi que l'opérateur de généralisation utilisé dans le cadre de l'apprentissage de grammaires catégorielles est strictement plus puissant que celui de "fusion d'états" utilisé en inférence grammaticale régulière, puisqu'il peut faire sortir de la classe des langages régulier. Son effet sur un automate à états finis reste toutefois interprétable, à condition d'étendre la définition des automates considérés. Cette piste donne lieu à la dernière partie de l'article et aboutit notamment à la caractérisation d'une nouvelle sous-classe de grammaires catégorielles, dont l'apprentissage est nettement moins coûteux que ce que les algorithmes de Kanazawa laissaient craindre.

Cet article propose en quelque sorte de rapprocher les connaissances et les techniques issues de deux communautés distinctes, en montrant qu'elles peuvent chacune tirer bénéfice des travaux de l'autre.

2 Automates à états finis et grammaires catégorielles

Dans cette section, nous introduisons les définitions nécessaires et nous explicitons les correspondances entre automates finis et grammaires catégorielles.

2.1 Automates à états finis et langages réguliers

Définition 1 (Automates à états finis (AF) et langages réguliers)

Un **automate à états finis** (AF par la suite) A est un quintuplet $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ où Q est l'ensemble fini des états de A , Σ est son vocabulaire fini, son état initial est $q_0 \in Q$ (nous nous restreignons ici aux automates avec un unique état initial) et $F \subseteq Q$ est l'ensemble de ses états finaux. Enfin, δ est la fonction de transition de A , définie de $Q \times \Sigma$ vers 2^Q .

Le langage $L(A)$ reconnu (ou engendré) par A est défini par : $L(A) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \cap F \neq \emptyset\}$, où δ^* est l'extension naturelle de δ à $Q \times \Sigma^*$ telle que : pour tout $a \in \Sigma$, tout $u \in \Sigma^*$ et tout $q \in Q$, $\delta^*(q, au) = \{\delta^*(q', u) \mid q' \in \delta(q, a)\}$.

L'ensemble de tous les langages reconnus par un AF est appelé l'ensemble des **langages réguliers**.

Exemple 1

La Figure 1 montre un AF A tel que $L(A) \stackrel{\text{def}}{=} a^+b^+$.

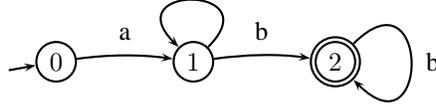


FIG. 1 – un automate à états finis

2.2 Grammaires formelles et grammaires catégorielles

Nous rappelons ici la définition classique des grammaires formelles ainsi que celle, moins connue, des grammaires catégorielles de type AB (en référence à leurs fondateurs, Adjukiewicz et Bar-Hillel), qui seront les seules considérées par la suite.

Définition 2 (Grammaires formelles et leur langage)

Une **grammaire formelle** (ou tout simplement une grammaire) G est un quadruplet $G = \langle \Sigma, N, P, S \rangle$ où Σ est le vocabulaire terminal fini de G , N son vocabulaire non terminal, lui aussi fini, $P \subset (\Sigma \cup N)^+ \times (\Sigma \cup N)^*$ son ensemble fini de règles de réécriture et $S \in N$ son axiome.

Le langage $L(G)$ reconnu (ou engendré) par G est défini par : $L(G) = \{w \in \Sigma^* \mid S \xrightarrow{*} w\}$ où $\xrightarrow{*}$ est la clôture réflexive et transitive de la relation définie par P .

Définition 3 (Catégories, grammaires catégorielles de type AB et leur langage)

Soit \mathcal{B} un ensemble au plus dénombrable de catégories de base, parmi lesquelles figure une catégorie distinguée $S \in \mathcal{B}$, appelée l'axiome. L'ensemble des **catégories fondées sur \mathcal{B}** , noté $Cat(\mathcal{B})$, est le plus petit ensemble tel que $\mathcal{B} \subset Cat(\mathcal{B})$ et pour tout $A, B \in Cat(\mathcal{B})$ on a : $A/B \in Cat(\mathcal{B})$ et $B \setminus A \in Cat(\mathcal{B})$.

Pour tout vocabulaire fini Σ (dont les membres seront appelés des mots) et pour tout ensemble \mathcal{B} de catégories de base ($S \in \mathcal{B}$), une **grammaire catégorielle** G est une relation finie sur $\Sigma \times Cat(\mathcal{B})$. On note $\langle v, A \rangle \in G$ l'affectation de la catégorie $A \in Cat(\mathcal{B})$ au mot $v \in \Sigma$. Une grammaire catégorielle de type AB (ou simplement une GC par la suite) est une grammaire catégorielle dont les règles de réécriture sont réduites aux deux schémas applicatifs suivants : $\forall A, B \in Cat(\mathcal{B})$

- FA (Forward Application) : $A/B \ B \rightarrow A$
- BA (Backward Application) : $B \ B \setminus A \rightarrow A$

Le langage $L(G)$ reconnu (ou engendré) par G est défini par : $L(G) = \{w = v_1 \dots v_n \in \Sigma^+ \mid \forall i \in \{1, \dots, n\}, \exists A_i \in Cat(\mathcal{B}) \text{ tel que } \langle v_i, A_i \rangle \in G \text{ et } A_1 \dots A_n \xrightarrow{*} S\}$, où $\xrightarrow{*}$ est la clôture réflexive et transitive de la relation définie par FA et BA.

Pour tout entier $k \geq 1$, l'ensemble des GCs qui affectent au plus k catégories distinctes à chacun des mots de leur vocabulaire est la classe des **GCs k -valuées**, et est notée \mathcal{G}_k . Les GCs de \mathcal{G}_1 sont aussi dites rigides.

La particularité principale des GCs est qu'elles sont *lexicalisées*, au sens où la totalité de l'information syntaxique est associée aux mots du vocabulaire (puisque les règles sont, elles, définies une fois pour toutes et invariables d'une grammaire à une autre).

Exemple 2

Les GCs ont surtout été utilisées pour la modélisation des langues naturelles. Soit par exemple $\mathcal{B} = \{S, T, NC\}$ (où T désigne la catégorie des “termes” et NC celle des “noms communs”), $\Sigma = \{Paul, dort, aime, un, chat\}$ et G la GC définie par : $G = \{\langle Paul, T \rangle, \langle dort, T \setminus S \rangle, \langle aime, (T \setminus S) / T \rangle, \langle chat, NC \rangle, \langle un, T / NC \rangle, \}$. G reconnaît des phrases comme “Paul dort”, “Paul aime un chat”, etc.

Définition 4 (FA-Structures, exemple structuré, langage des structures)

Une **FA-structure** (FA pour “Foncteur-Argument”) sur un vocabulaire Σ est un arbre binaire dont les feuilles sont étiquetées par des éléments de Σ et dont chaque noeud est étiqueté soit par BA soit par FA . L'ensemble des FA-structures sur Σ est noté Σ^F .

Pour toute GC $G \subset \Sigma \times \text{Cat}(\mathcal{B})$, un **exemple structuré** pour G est un élément de Σ^F qui est obtenu à partir d'un arbre d'analyse syntaxique produit par G pour une phrase $w \in L(G)$, en effaçant dans cet arbre toutes les catégories de $\text{Cat}(\mathcal{B})$. Pour toute GC G , le **langage des structures** de G , noté $FL(G)$ est l'ensemble de ses exemples structurés.

2.3 Des automates aux grammaires catégorielles et inversement

Les GCs peuvent engendrer tous les langages algébriques sans ϵ (où ϵ désigne le mot vide) (Bar-Hillel *et al.*, 1960). Elles peuvent donc aussi engendrer tous les langages réguliers sans ϵ . Les correspondances entre AFs et GCs sont faciles à définir.

Définition 5 (GCs régulières)

Nous appelons **GC régulière** une CG $G \subset \Sigma \times \text{Cat}(\mathcal{B})$ qui ne contient que des affectations de la forme $\langle v, A \rangle$ ou $\langle v, A/B \rangle$ avec $v \in \Sigma$ et $A, B \in \mathcal{B}$. L'ensemble des GCs régulières est noté \mathcal{G}_r .

Propriété 1 (Transformation d'un AF en GC)

Soit $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AF. Soit $\mathcal{B} = (Q \setminus \{q_0\}) \cup \{S\}$ (où $S \notin Q$). Il est possible de définir une GC régulière $G \subset \Sigma \times \text{Cat}(\mathcal{B})$ telle que $L(G) = L(A) \setminus \{\epsilon\}$.

Preuve 1 (Preuve de la Propriété 1)

Cette propriété découle de la transformation classique d'un AF en une grammaire linéaire à gauche $G_1 = \langle \Sigma, Q, P_1, S \rangle$, que l'on peut ensuite transformer en une GC $G \in \mathcal{G}_r$. On remplace q_0 par S puis $\forall a \in \Sigma$ et $\forall q, q' \in Q$ tels que $q' \in \delta(q, a)$ on fait :

- si $q' \in F$ alors
 - $q \rightarrow a$ est une règle de G_1 (élément de P_1) et $\langle a, q \rangle \in G$;
 - si $\exists u \in \Sigma$ tel que $\exists q'' \in \delta(q', u)$ alors $q \rightarrow aq'$ est une règle de G_1 et $\langle a, q/q' \rangle \in G$;
 - sinon $q \rightarrow aq'$ est une règle de G_1 et $\langle a, q/q' \rangle \in G$.

$\epsilon \in L(A)$ si et seulement si $q_0 \in F$. Cette situation amène à ajouter $S \rightarrow \epsilon$ dans P_1 . Mais, dans une GC, il est impossible d'affecter une catégorie à ϵ . Cette règle n'a donc pas de contrepartie dans G et nous avons : $L(A) \setminus \{\epsilon\} = L(G_1) \setminus \{\epsilon\} = L(G)$. \square

Exemple 3

En appliquant la propriété précédente à l'AF de l'Exemple 1, nous obtenons les règles de G_1 suivantes (où l'état i est associé à un non-terminal noté q_i , avec $q_0 = S$) :

$S \longrightarrow aq_1, q_1 \longrightarrow aq_1, q_1 \longrightarrow b, q_1 \longrightarrow bq_2, q_2 \longrightarrow b, q_2 \longrightarrow bq_2$. La GC obtenue est : $G = \{\langle a, S/q_1 \rangle, \langle a, q_1/q_1 \rangle, \langle b, q_1 \rangle, \langle b, q_1/q_2 \rangle, \langle b, q_2 \rangle, \langle b, q_2/q_2 \rangle\}$.

Les AF peuvent donc simplement être *lexicalisés*, sous la forme d’une GC. Remarquons que, dans les GCs régulières que nous avons définies, seul l’opérateur / est utilisé dans les catégories affectées aux mots du vocabulaire, et seul le schéma FA est utile (on aurait pu aussi se restreindre à l’opérateur \ et au schéma BA en transformant l’AF en une grammaire linéaire à gauche). En fait, la transformation précédente ne préserve pas seulement le langage final, mais aussi les structures d’analyse, qui sont des peignes. Ainsi, les exemples structurés, tels que définis dans la Définition 4 sont en quelque sorte disponibles “gratuitement” à partir des phrases.

Exemple 4

La Figure 2 montre deux arbres d’analyse syntaxique produits par la CG de l’Exemple 3, et (à droite) les exemples structurés correspondant.

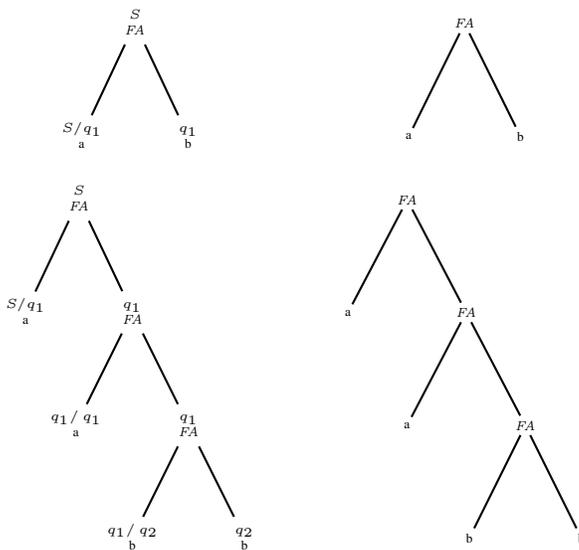


FIG. 2 – Deux analyses syntaxiques et les exemples structurés correspondant

Propriété 2 (Transformation d’une GC régulière en un AF)

Toute GC régulière $G \subset \Sigma \times \text{Cat}(\mathcal{B})$ peut être transformée en un AF $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ reconnaissant le même langage (sans ϵ).

Preuve 2 (Schéma de preuve de la Propriété 2)

Cette transformation est l’inverse de celle exposée dans la Propriété 1 : le seul point notable est qu’il est plus facile d’ajouter un unique état final F_A dans A . Soit donc $Q = \mathcal{B} \cup \{F_A\}$ avec $F_A \notin \mathcal{B}, q_0 = S$ et $F = \{F_A\}$. Chaque affectation $\langle a, U/V \rangle \in G$ correspond à une transition étiquetée par a entre les états U et V dans A (soit encore :

$\delta(U, a) = V$) et chaque affectation $\langle a, U \rangle \in G$ à une transition étiquetée par a entre U et F_A ($\delta(U, a) = F_A$).□

Exemple 5

L'AF obtenu en appliquant cette opération à la GC de l'Exemple 3 est donné en Figure 3. Il ne coïncide pas exactement avec celui de l'Exemple 1 à cause de l'état final ajouté à ceux provenant des catégories de base. Le résultat est, de ce fait, non déterministe.

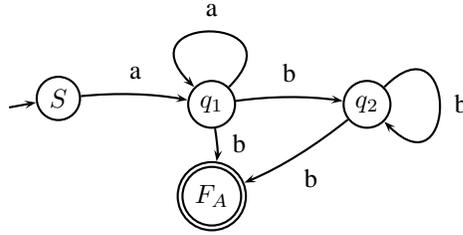


FIG. 3 – AF obtenu à partir d'une GC

Remarque 1

Les Propriétés 1 et 2 ne signifient pas que seules les GCs régulières génèrent des langages réguliers. Celle de l'Exemple 2 n'est pas régulière au sens de la Définition 5 mais elle génère un langage fini (donc régulier). Mais elle ne produit pas que des peignes.

Propriété 3 (Langage associé à une catégorie ou à un état)

Soit G une GC régulière et A l'AF obtenu à partir de G . Alors, pour toute catégorie de base $q \in \mathcal{B}$ (correspondant à un état non final $q \in Q$ de A), nous avons deux façons distinctes de caractériser le langage $L(q)$ associé à q :

$L(q) = \{w = v_1 \dots v_n \in \Sigma^+ \mid \forall i \in \{1, \dots, n\} \exists A_i \in \text{Cat}(\mathcal{B}) \text{ tel que } \langle v_i, A_i \rangle \in G \text{ et } A_1 \dots A_n \rightarrow^* q\}$ et $L(q) = \{w \in \Sigma^+ \mid F_A \in \delta^+(q, w)\}$.

Preuve 3 (Schéma de preuve de la Propriété 3)

Cette propriété est une conséquence triviale des Propriétés 1 et 2, où q remplace S .□

La seconde définition de $L(q)$ n'est correcte que parce que nous savons que $F_A \neq q$ est l'unique état final de A . Ainsi, les successions de mots (ou chaînes) auxquelles G associe la catégorie q sont celles qui correspondent dans A à un chemin qui part de l'état q et aboutit à l'état F_A . Bien sûr, si $q = q_0 = S$, on retrouve $L(S) = L(A) = L(G)$.

Exemple 6

Dans l'AF de la Figure 3, $L(q_1) = a^*b^+$ et $L(q_2) = b^+$.

3 Inférence de GCs par exemples positifs

L'étude de l'apprenabilité des GCs par exemples positifs seuls au sens de Gold (Gold, 1967) a donné lieu à de nombreux travaux récents, initiés par Kanazawa (Kanazawa,

1998). Maintenant que nous disposons d'un mécanisme de traduction d'une sous-classe des GCs (les GCs régulières) en AFs, il est naturel de se demander si des résultats concernant l'apprenabilité de ces GCs peuvent se traduire en résultats d'apprenabilité de langages réguliers.

3.1 Modèle de Gold

Définition 6 (Critère d'apprenabilité)

Soit \mathcal{G} un ensemble de grammaires (par la suite, \mathcal{G} sera une sous-classe de l'ensemble des GCs) sur un alphabet Σ et soit une fonction qui associe un langage à chaque grammaire. Cette fonction sera soit le langage des chaînes $L : \mathcal{G} \rightarrow \text{pow}(\Sigma^*)$ (cf. Définition 3) soit le langage des structures $FL : \mathcal{G} \rightarrow \text{pow}(\Sigma^F)$ (cf. Définition 4).

Soit ϕ une fonction qui à tout échantillon fini de phrases de Σ^* (resp. d'exemples structurés de Σ^F) associe une grammaire de \mathcal{G} . On dit que cette fonction **converge** vers $G \in \mathcal{G}$ sur un échantillon $\langle s_i \rangle_{i \in \mathbb{N}}$ d'éléments de Σ^* (resp. de Σ^F) si $G_i = \phi(\langle s_0, \dots, s_i \rangle)$ est défini et égal à G pour un nombre fini de valeurs de $i \in \mathbb{N}$ ou, ce qui revient au même, si $\exists n_0 \in \mathbb{N}$ tel que pour tout $i \geq n_0$, G_i est défini et égal à G .

On dit que ϕ **apprend** la classe \mathcal{G} par exemples positifs (resp. par exemples structurés positifs) si pour tout langage L de $L(\mathcal{G}) = \{L(G) | G \in \mathcal{G}\}$ (resp. de $FL(\mathcal{G}) = \{FL(G) | G \in \mathcal{G}\}$) et pour toute séquence $\langle s_i \rangle_{i \in \mathbb{N}}$ qui énumère L , c'est-à-dire telle que $\{s_i | i \in \mathbb{N}\} = L$, il existe $G \in \mathcal{G}$ telle que $L = L(G)$ (resp. $L = FL(G)$) et ϕ converge vers G sur $\langle s_i \rangle_{i \in \mathbb{N}}$.

\mathcal{G} est **apprenable** s'il existe une fonction ϕ calculable qui apprend \mathcal{G} .

Théorème 1 (Apprenabilité de \mathcal{G}_k (Kanazawa, 1998))

Pour tout entier $k \geq 1$, la classe des GCs k -valuées \mathcal{G}_k est apprenable par exemples positifs et par exemples structurés positifs.

3.2 Apprentissage de GCs et inférence grammaticale régulière

Dans cette section, nous traduisons les résultats d'apprenabilité de Kanazawa dans la classe des GCs régulières, puis, nous rapprochons deux résultats connus.

Théorème 2 (Apprenabilité des GCs régulières k -valuées)

Pour tout entier $k \geq 1$, la classe des GCs régulières k -valuées $\mathcal{G}_k \cap \mathcal{G}_r$ est apprenable par exemples positifs et par exemples structurés positifs.

Preuve 4 (Schéma de preuve du Théorème 2)

Ce théorème est une conséquence du Théorème 1, restreint à la classe \mathcal{G}_r . Tout algorithme d'apprentissage de \mathcal{G}_k à partir de structures peut être adapté pour devenir un algorithme d'apprentissage de $\mathcal{G}_k \cap \mathcal{G}_r$ à partir de chaînes, en associant des peignes avec noeuds FA aux chaînes et en ne conservant que les sorties de la fonction d'apprentissage qui sont isomorphes à des grammaires régulières (ce qui est décidable). \square

Bien sûr, $\bigcup_{k \geq 1} \{L(G) | G \in \mathcal{G}_k \cap \mathcal{G}_r\}$ contient tous les langages réguliers. Mais le principal intérêt de cette classe est que, contrairement au cas général, elle tout aussi

facile à apprendre à partir de chaînes qu'à partir d'exemples structurés. Cette propriété sera étendue au delà des GCs régulières dans la section suivante.

Cependant le résultat d'apprenabilité lui-même n'est pas très surprenant. En effet, pour tout entier k , les AFs qui traduisent les GCs de $\mathcal{G}_k \cap \mathcal{G}_r$ (suivant le processus décrit dans la preuve de la Propriété 2) ont au plus k transitions étiquetées par le même mot du vocabulaire. Or, le nombre d'AFs vérifiant cette propriété est fini. L'apprentissage par exemples positifs d'un ensemble fini est toujours possible.

Néanmoins, cette classe d'automates a des caractéristiques originales. Elle inclut tout aussi bien des AFs déterministes que des AFs non-déterministes. Et, contrairement aux classes habituellement considérées en inférence grammaticale régulière, elle semble bien adaptée aux grands alphabets (à condition que k soit petit).

Pour compléter le rapprochement que nous avons commencé à établir entre AFs et GCs, nous rapprochons dans ce qui suit deux résultats connus.

Définition 7 (AFs 0-réversibles (Angluin, 1982))

Un AF est dit 0-réversible si et seulement si il est déterministe, et l'AF obtenu en inversant le sens de ses transitions, et en échangeant les rôles des états initiaux et finaux est déterministe. La classe des AFs 0-réversibles est apprenable par exemples positifs.

Définition 8 (CG réversible (Besombes & Marion, 2004))

Une GC est dite réversible si elle ne contient pas deux affectations de catégories pour un même mot du vocabulaire, qui ne sont distinctes que par une seule catégorie de base. La classe des GCs réversibles est apprenable à partir d'exemples structurés et de chaînes.

Théorème 3 (Equivalence entre ces deux notions de réversibilité)

Soit G une GC régulière et A l'AF obtenu à partir de G . A est 0-réversible dans le sens de la Définition 7 si et seulement si G est réversible dans le sens de la Définition 8.

Preuve 5 (Preuve du Théorème 3)

Par construction, A n'a qu'un seul état initial et aucune transition étiquetée par ϵ . Pour que A soit déterministe, il suffit donc qu'il n'existe pas deux transitions avec la même étiquette au départ d'un même état. Cette condition se traduit dans G par : $\forall a \in \Sigma$

- $\forall Q_1, Q_2, Q_3 \in \mathcal{B} : \langle a, Q_1/Q_2 \rangle \in G \text{ et } \langle a, Q_1/Q_3 \rangle \in G \Leftrightarrow Q_2 = Q_3.$
- $\forall Q_1, Q_2 \in \mathcal{B} : \langle a, Q_1 \rangle \in G \text{ et } \langle a, Q_1/Q_2 \rangle \in G \Leftrightarrow Q_2 = F_A$ ($\langle a, Q_1 \rangle$ joue en quelque sorte le rôle de $\langle a, Q_1/F_A \rangle$ et aucune transition ne part de F_A);

De même, A n'a qu'un seul état final donc la condition pour que son inverse soit déterministe se traduit dans G de la façon suivante : $\forall a \in \Sigma$

- $\forall Q_1, Q_2 \in \mathcal{B} : \langle a, Q_1 \rangle \in G \text{ et } \langle a, Q_2 \rangle \in G \Leftrightarrow Q_1 = Q_2$
- $\forall Q_1, Q_2, Q_3 \in \mathcal{B} : \langle a, Q_1/Q_2 \rangle \in G \text{ et } \langle a, Q_3/Q_2 \rangle \in G \Leftrightarrow Q_1 = Q_3.$

Pour les GCs régulières, ces conditions coïncident avec celles de la Définition 8. \square

Ainsi, le résultat d'apprenabilité de la classe des AFs 0-réversibles à partir de chaînes (Angluin, 1982) se déduit du résultat d'apprenabilité de la classe des GCs régulières réversibles par exemples structurés, qui lui-même découle de (Besombes & Marion, 2004). Là encore, ce résultat n'est pas très surprenant : (Besombes & Marion, 2004) se sont inspirés des grammaires algébriques réversibles de (Sakakibara, 1992) qui, lui-même, s'était inspiré de (Angluin, 1982).

3.3 Algorithme d'apprentissage

Les résultats d'apprenabilité du théorème 1 ne sont pas uniquement théoriques : ils s'accompagnent de la définition d'une fonction d'apprentissage originale. Elle est fondée sur un algorithme que nous appellerons BP, en hommage à ses inventeurs (Buszkowski & Penn, 1990), qui est capable, pour tout entier k et tout ensemble d'exemples structurés D , d'identifier l'ensemble des GCs k -valuées sans catégorie inutile (cf. définition 12 plus loin) compatibles avec D . Nous rappelons ici le principe de cet algorithme, en l'illustrant sur un exemple où les éléments de D sont des peignes avec FA pour noeuds internes. Pour tout élément de D , les premières étapes de BP consistent à :

1. introduire l'étiquette S à la racine de chaque exemple structuré ;
2. introduire une variable distincte x_i à chaque noeud argument (c'est-à-dire, au fils gauche de chaque noeud BA , et au fils droit de chaque noeud FA) ;
3. introduire à chaque autre noeud la catégorie qui rend possible l'application des schémas FA et BA qui étiquettent ces noeuds.

La GC définie en récoltant les affectations de catégories aux feuilles des éléments de D à la fin de ces étapes est appelée **forme générale** de D et notée $FG(D)$.

Exemple 7

Soit D défini comme l'ensemble des deux exemples structurés de l'Exemple 4. Les étapes 1 à 3 décrites précédemment aboutissent au résultat donné par la Figure 4, et $FG(D)$ est alors définie comme suit :

- a : $S/x_1, S/x_4, x_4/x_3$;
- b : $x_1, x_3/x_2, x_2$.

L'AF correspondant est donné en Figure 5 : il est presque identique à ce que, en inférence grammaticale, on appelle **automate canonique minimal** reconnaissant $\{ab, aabb\}$. La seule différence est que notre AF a un unique état initial et un unique état final.

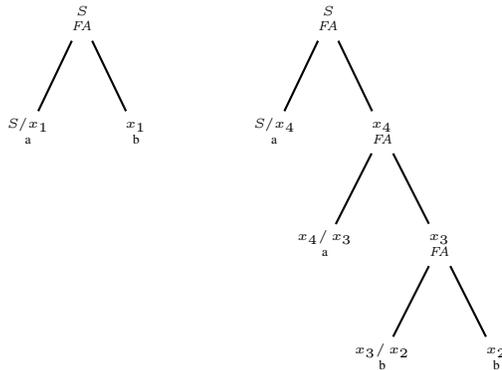


FIG. 4 – Résultat de l'application des 3 premières étapes de BP

La suite de l'algorithme BP consiste à chercher des *substitutions unificatrices* applicables sur $FG(D)$. On précise donc tout d'abord ce que sont ces substitutions.

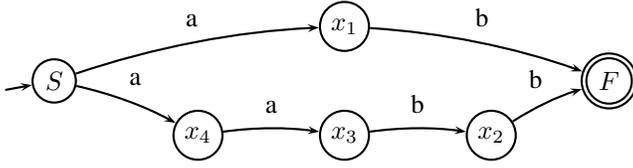


FIG. 5 – l'AF correspondant à $FG(D)$

Définition 9 (Catégories avec variables et substitutions)

Soit χ un ensemble infini dénombrable de variables et soit $\mathcal{B} = \chi \cup \{S\}$. Une **substitution** est une fonction $\sigma : \chi \rightarrow \text{Cat}(\mathcal{B})$ qui transforme une variable en une catégorie (par défaut, elle vaut l'identité sur χ). Une substitution est étendue par morphisme à une fonction de $\text{Cat}(\mathcal{B})$ dans $\text{Cat}(\mathcal{B})$ de la manière suivante : (i) $\sigma(S) = S$, (ii) $\sigma(A/B) = \sigma(A)/\sigma(B)$ et (iii) $\sigma(A \setminus B) = \sigma(A) \setminus \sigma(B)$ pour tout $A, B \in \text{Cat}(\mathcal{B})$. De même, une substitution peut être étendue à une GC quelconque $G : \sigma(G) = \{\langle v, \sigma(A) \rangle \mid \langle v, A \rangle \in G\}$. Pour toute GC G , une **substitution unificatrice** de G est une substitution qui unifie des catégories affectées à un même mot du vocabulaire de G .

Propriété 4 (Propriété Fondamentale (Buszkowski & Penn, 1990))

Pour toute GC G , les propriétés suivantes sont équivalentes : (i) $D \subseteq FL(G)$ et (ii) $\exists \sigma$ telle que $\sigma[GF(D)] \subseteq G$.

En d'autres termes, les GCs compatibles avec un ensemble d'exemples structurés D sont celles qui incluent une substitution de $FG(D)$. Quand il cible des GCs k -valuées, l'étape 4 de l'algorithme BP consiste donc à chercher toutes les substitutions unificatrices de $FG(D)$ qui sont dans \mathcal{G}_k . Le résultat de BP est donc un ensemble de GCs. Si D est constitué d'exemples structurés et $k = 1$, la grammaire qui a produit D , si elle existe, est unique et isomorphe, à la limite, au résultat de BP : BP est alors un algorithme d'apprentissage efficace. Si $k > 1$, le résultat de BP n'est en général pas réduit à un singleton. Pour obtenir une fonction d'apprentissage au sens de Gold, il faut donc être capable de faire un choix parmi ses éléments. Cette étape nécessite de réaliser des tests d'inclusion et est très coûteuse en temps de calcul. Nous ne la détaillerons pas ici.

Appliquer une substitution à une GC est une opération de généralisation. En effet, on a la propriété suivante (Buszkowski & Penn, 1990) : $\sigma(G_1) \subseteq G_2 \implies FL(G_1) \subseteq FL(G_2)$, qui implique que : $FL(G) \subseteq FL(\sigma(G))$. De même : $L(G) \subseteq L(\sigma(G))$. Cette opération a-t-elle un lien avec celle de fusion d'états, qui est utilisée en inférence grammaticale régulière (Angluin, 1982; Oncina & Garcia, 1992; Dupont *et al.*, 1994) ? C'est ce que nous allons voir maintenant. Dans le cas d'un ensemble D constitué de peignes avec noeuds internes FA , $FG(D)$ est toujours une GC régulière. Donc, pour unifier deux catégories de $FG(D)$, seuls deux cas peuvent se produire :

- des conditions de la forme $\sigma(x_i) = \sigma(x_j) = x_j$ pour x_i et x_j dans χ spécifient précisément une fusion des états x_i et x_j dans l'AF correspondant à $FG(D)$.
- des conditions de la forme $\sigma(x_i) = x_j/x_k$, pour $x_i \in \chi$ et $x_j, x_k \in \chi \cup \{S\}$ sont, au premier abord, plus difficiles à interpréter. Elles signifient en fait deux choses :
 - l'état x_i est renommé en x_j/x_k ;

- toutes les chaînes auxquelles $FG(D)$ affectait la catégorie x_i peuvent désormais être utilisées comme “transition” pour passer de l’état x_j à l’état x_k .

Exemple 8

Définissons une substitution unifi catrice σ pour la $FG(D)$ obtenue dans l’Exemple 7 comme suit : $\sigma(x_4) = \sigma(x_1) = x_3/x_2$ (et σ est l’identité partout ailleurs). La GC $\sigma(GF(D))$ est alors définie par :

- $a : S/(x_3/x_2), (x_3/x_2)/x_3$;
- $b : x_3/x_2, x_2$.

Cette GC n’est plus régulière, à cause des catégories affectées à a . Néanmoins, on peut encore lui associer un automate en intégrant à celui-ci une “transition récursive”, c’est-à-dire une transition étiquetée non plus par un élément du vocabulaire, mais par un état (ici, x_3/x_2). L’automate obtenu est celui de la Figure 6.

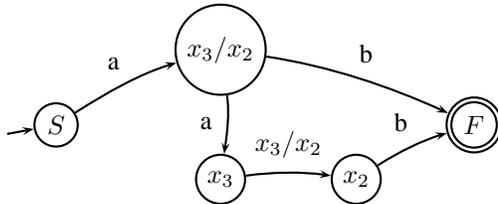


FIG. 6 – L’automate généralisé correspondant à $\sigma(FG(D))$

Dans cet automate, les états nommés x_1 et x_4 dans l’AF de la Figure 5 ont été fusionnés, sous l’effet de la condition $\sigma(x_4) = \sigma(x_1)$. La condition $\sigma(x_1) = x_3/x_2$ a, elle, eu pour effet de renommer cet état en x_3/x_2 , et de remplacer la transition étiquetée par b entre x_3 et x_2 par une transition étiquetée par x_3/x_2 : c’est ce que nous appelons une transition récursive. Pour la franchir, il faut produire une chaîne de catégorie x_3/x_2 c’est-à-dire, d’après la Propriété 3, une chaîne qui correspond à un chemin qui part de l’état x_3/x_2 et aboutit à l’état final F . Le premier exemple d’un tel chemin, c’est évidemment la transition étiquetée par b qui relie x_3/x_2 à F . Mais, partant de l’état x_3/x_2 , il est aussi possible d’emprunter d’abord la transition étiquetée par a qui mène à x_3 , avant de franchir une nouvelle fois la transition récursive. Une pile (implicite) est nécessaire pour enregistrer tous les appels récursifs successifs issus du franchissement de cette transition. Le langage reconnu par cet automate généralisé n’est autre que $a^n b^n$ qui, bien sûr, n’est pas régulier. On peut rapprocher ce dispositif de celui des Réseaux de Transitions Récursifs ou RTRs (Woods, 1970), mais réduit à un seul automate (dans les RTRs, il y a autant d’automates que de symboles non terminaux).

L’arbre d’analyse syntaxique associé à la chaîne $aaabbb$ par $\sigma(FG(D))$ est donné Figure 7. Cet arbre n’est plus un peigne. Pour comprendre sa construction, le mieux est de se reporter à l’arbre d’analyse syntaxique associé à $aabb$ par $FG(D)$, en Figure 4. Dans ce dernier arbre, on voit que x_4 étiquette un noeud interne, tandis que x_3/x_2 étiquette une feuille. La condition $\sigma(x_4) = x_3/x_2$ ouvre donc la possibilité de substituer le sous-arbre de racine x_4 à la place de la feuille x_3/x_2 . Cette opération, qui coïncide

exactement avec ce qui est appelé une adjonction dans le formalisme des Tree Adjoining Grammars (Joshi & Schabes, 1997), produit comme résultat l'arbre de la Figure 7.

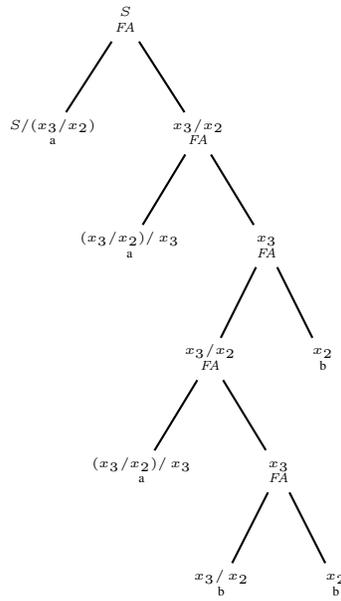


FIG. 7 – Arbre d'analyse syntaxique de $aaabbb$ par $\sigma(FG(D))$

4 Apprentissage de langages à partir de peignes

L'Exemple 8 suggère qu'il est possible de produire de vrais arbres par adjonctions de peignes, et de représenter les langages algébriques par des automates généralisés. Cette section est consacrée à la formalisation de ces idées, et à leur exploitation pour améliorer l'algorithme d'apprentissage de GCs à partir de chaînes de Kanazawa.

4.1 Automates récursifs et leur expressivité

Définition 10 (Automate récursif)

Un **automate récursif** (AR par la suite) R est un quintuplet $R = \langle Q, \Sigma, \gamma, q_0, F \rangle$ où Q est l'ensemble fini des états de R , Σ son vocabulaire fini, son unique état initial est $q_0 \in Q$ et $F \in Q$ son (unique) état final. γ est la fonction de transition de R , définie de $Q \times (\Sigma \cup Q)$ vers 2^Q .

La seule différence importante entre les ARs et les AFs est que, dans un AR, les transitions peuvent être étiquetées soit par un élément de Σ , soit par un élément de Q . Dans ce dernier cas, on parle de **transition récursive**. Pour franchir une transition récursive, il faut produire un élément du langage de l'état qui étiquette la transition.

Nous nous restreignons ici aux ARs qui ont un unique état initial et un unique état final, mais cela ne restreint pas l'expressivité du modèle. Comme nous nous intéresserons aux langages sans ϵ , on peut supposer de plus que $F \neq q_0$.

Définition 11 (Language Reconnu par un AR)

Le langage $L(R)$ reconnu (ou engendré) par un AR $R = \langle Q, \Sigma, \gamma, q_0, F \rangle$ est le plus petit ensemble défini par : $L(R) = \{w \in \Sigma^+ \mid F \in \gamma^+(q_0, w)\}$, où γ^+ est l'extension naturelle de γ à $Q \times \Sigma^+$. Pour tout $u \in \Sigma^+$, tout $v \in \Sigma^*$ et tout $q \in Q$, $\gamma^+(q, uv)$ est défini comme le plus petit sous-ensemble contenant $\{\gamma^*(q', v) \mid q' \in \gamma(q, u)\}$ si $u \in \Sigma$ et $\{\gamma^*(q', v) \mid \exists t \in Q \text{ tel que } q' \in \gamma(q, t) \text{ et } u \in L(t)\}$ sinon, où $L(t)$ est le langage de l'état t , défini comme en Définition 3 en remplaçant δ par γ . Cette définition de $L(G)$ est récursive : $L(G)$ est défini comme un plus petit point fixe, quand il existe.

Une transition récursive étiquetée par un état $q \in Q$ sera dite *vraiment récursive* s'il existe un chemin qui mène de l'état q à F en passant par cette transition. Les transitions *vraiment récursives* sont celles qui permettent des adjonctions et produisent donc des structures qui ne sont pas nécessairement des peignes.

Théorème 4 (Des GCs unidirectionnelles aux ARs)

Une GC unidirectionnelle n'affecte que des catégories qui sont soit de base, soit construites avec l'opérateur / uniquement. L'ensemble des GCs unidirectionnelles sera noté $\mathcal{G}_/$. Pour toute GC dans $G \in \mathcal{G}_/$, on peut construire un AR fortement équivalent à G , c'est-à-dire produisant les mêmes structures.

Preuve 6 (Schéma de la preuve du Théorème 4)

Il est connu depuis longtemps que toute GC $G \subset \Sigma \times \text{Cat}(\mathcal{B})$ peut être transformée en une grammaire algébrique $H = \langle \Sigma, N, P, S \rangle$ sous forme normale de Chomsky fortement équivalent à G . H est construite de la façon suivante : N est l'ensemble de toutes les sous-catégories d'une catégorie présente dans les affectations de G (une catégorie est une sous-catégorie d'elle-même) et P contient toutes les règles de la forme $A \rightarrow v$ pour tout $\langle v, A \rangle \in G$ et toutes les règles de la forme $A \rightarrow A/B$ pour toute catégorie A et B dans N (pour les GC unidirectionnelles, cela suffit). Pour construire un AR à partir de ces règles, il suffit de procéder exactement comme avec les règles régulières utilisées dans la Preuve de la Propriété 2. \square

Corollaire 1 (Corollaire du Théorème 4)

Les grammaires de $\mathcal{G}_/$ peuvent produire tous les langages algébriques sans ϵ (Bar-Hillel et al., 1960). Donc, les ARs peuvent également produire tous ces langages.

Exemple 9

La CG de $\mathcal{G}_/$ classique qui reconnaît $a^n b^n$, $n \geq 1$, est : $\{\langle a, S/B \rangle, \langle a, (S/B)/S \rangle, \langle b, B \rangle\}$. L'AR correspondant (distinct de celui de l'Exemple 8) est donné en Figure 8. Cet AR peut être simplifié : les transitions récursives qui ne sont pas vraiment récursives peuvent être lexicalisées. Ici, on peut effacer l'état $(S/B)/S$ et remplacer la transition récursive qui y fait référence entre S/B et S par a . Ce n'est possible pour aucun autre état.

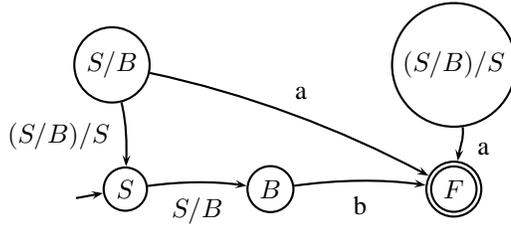


FIG. 8 – Un autre AR reconnaissant $a^n b^n$

Les ARs produisent les mêmes structures que les GCs unidirectionnelles, c'est-à-dire ne faisant appel qu'au schéma FA . On espère donc, comme dans l'Exemple 8, les apprendre à partir de peignes uniquement. Mais l'AR de la Figure 8 n'appartient à aucun espace de recherche qui partirait d'un ensemble de peignes, parce qu'il comprend des états qui ne sont pas accessibles à partir de l'état initial. On devra donc contraindre un peu plus nos ARs (ou nos GCs unidirectionnelles) pour espérer les apprendre ainsi.

4.2 Apprentissage de GCs unidirectionnelles à partir de peignes

Quand les données de départ sont des chaînes uniquement, la stratégie d'apprentissage employée par Kanazawa (Kanazawa, 1998) consiste à générer toutes les structures possibles compatibles avec ces chaînes, avant de lancer l'algorithme d'inférence à partir d'exemples structurés décrit en section 3.3. Nous proposons une nouvelle stratégie bien moins coûteuse, adaptée à de nouvelles sous-classes de GCs.

Définition 12 (Nouvelles sous-classes de GCs)

Une GC G est dite sans catégorie inutile si toute affectation d'une catégorie à un mot du vocabulaire dans G est utilisée au moins une fois dans l'analyse syntaxique d'un élément de $L(G)$ (Kanazawa, 1998). Soit :

$\mathcal{G}_k^{FA} = \{\sigma(G) | G \in \mathcal{G}_k \cap \mathcal{G}_/ \text{ et } G \text{ est sans catégorie inutile et } \sigma \text{ est une substitution unifi catrice pour } G\}$.

Bien sûr, pour tout $k \geq 1$, $\mathcal{G}_k^{FA} \subset \mathcal{G}_k \cap \mathcal{G}_/$. De plus, $\bigcup_{k \geq 1} \{L(G) | G \in \mathcal{G}_k^{FA}\}$ contient tous les langages réguliers (cf. commentaires du Théorème 2, sachant que $\sigma = Id$ peut être considérée comme un cas particulier de substitution unifi catrice) et certains langages algébriques (voir Exemple 8). Mais elle ne semble pas contenir tous les langages algébriques et nous ne savons pas encore caractériser précisément son expressivité. Le problème est qu'un état inutile parce que non accessible dans un AF (correspondant à une catégorie inutile dans une GC) peut devenir utile bien que toujours inaccessible après qu'une substitution ait été appliquée à l'AF pour le transformer en un AR. Le théorème suivant explicite la propriété fondamentale des éléments de \mathcal{G}_k^{FA} .

Théorème 5

Pour tout $k \geq 1$ et toute GC $G \in \mathcal{G}_k^{FA}$, il existe un ensemble fini $D \subset FL(G)$ constitué uniquement de peignes avec noeuds internes FA et il existe τ , une substitution unifi catrice pour $FG(D)$, tels que $G = \tau(FG(D))$.

Preuve 7 (Preuve du Théorème 5)

Pour tout $G \in \mathcal{G}_k^{FA}$, par définition, $\exists G' \in \mathcal{G}_k \cap \mathcal{G}_r$ sans catégorie inutile et $\exists \sigma$ une substitution unificatrice pour G telles que $G = \sigma(G')$. Nous savons, d'après le Théorème 2 que $\mathcal{G}_k \cap \mathcal{G}_r$ est apprenable par exemples structurés. Soit D un ensemble caractéristique d'exemples structurés pour G' (voir (Kanazawa, 1998)). G' est régulière, donc D n'est constitué que de peignes avec noeuds internes FA. G' est sans catégorie inutile, donc elle appartient au résultat de l'algorithme BP appliqué aux données k et D (Kanazawa, 1998). Cela signifie qu'il existe une substitution unificatrice ρ pour $FG(D)$ telle que $G' = \rho(FG(D))$. On a donc $G = \sigma(G') = \sigma(\rho(FG(D)))$. On prend donc $\tau = \sigma \circ \rho$. \square

Le Théorème 5 signifie que les membres de \mathcal{G}_k^{FA} ont un ensemble caractéristique qui n'est constitué que de peignes avec noeuds internes FA, et qu'ils appartiennent donc au résultat de l'algorithme BP appliqué à cet ensemble. Cela suggère un nouvel algorithme pour apprendre la classe \mathcal{G}_k^{FA} à partir de chaînes (voir l'Algorithme 1), qui n'a pas besoin de générer toutes les structures possibles associées à ces chaînes.

Algorithm 1 algorithme qui infère des GCs dans \mathcal{G}_k^{FA} qui reconnaissent $\langle s_0, \dots, s_i \rangle$

Require: $\langle s_0, \dots, s_i \rangle$ où $\forall i, s_i \in \Sigma^+$ et k

- 1: $j \leftarrow 0$
- 2: **repeat**
- 3: $C_j \leftarrow \{s_0, \dots, s_j\} \setminus \setminus$ essayer C_j comme ensemble caractéristique
- 4: associer une structure de peignes avec noeuds FA à tous les éléments de C_j
- 5: appliquer l'algorithme BP à cet ensemble pour obtenir l'ensemble $R_{j,k} \subset \mathcal{G}_k$ des GCs sans catégorie inutile compatibles avec lui
- 6: supprimer les éléments de $R_{j,k}$ dont le langage de chaînes ne contient pas $\{s_{j+1}, \dots, s_i\}$
- 7: $j \leftarrow j + 1$
- 8: **until** $(j = i + 1)$ OR $(R_{j,k} \neq \emptyset)$

Ensure: $R_{j,k}$: un ensemble de GCs de \mathcal{G}_k^{FA} qui reconnaissent $\langle s_0, \dots, s_i \rangle$

Les GCs de \mathcal{G}_k^{FA} sont au plus k -valuées. Mais la valeur de k requise par l'Algorithme 1 peut être plus grande que celle requise par l'algorithme BP. Les GCs de \mathcal{G}_k^{FA} sont en quelque sorte sous une certaine forme normale : elles ne produisent que des peignes ou des adjonctions de peignes. Le Théorème 5 assure que, pour toute grammaire $G \in \mathcal{G}_k^{FA}$ produisant $L(G) = \langle s_i \rangle_{i \in \mathbb{N}}$, l'Algorithme 1 contient, à la limite, G parmi ses résultats.

5 Conclusion

Cette étude est un premier pas pour intégrer dans un même cadre les travaux réalisés en inférence grammaticale de langages réguliers et ceux réalisés en apprentissage de grammaires catégorielles. Les premiers bénéfices de ce rapprochement sont l'obtention, sans beaucoup d'efforts, de résultats originaux, et surtout une meilleure compréhension de la nature des opérations de généralisation utilisées dans chacune des deux approches. Une autre conséquence, sans doute moins attendue, de ce travail, est la définition d'une

nouvelle classe d'automates qui étend naturellement celle des automates finis tout en entretenant des liens forts avec les grammaires catégorielles unidirectionnelles.

Ce rapprochement montre surtout que le domaine de l'inférence grammaticale de langages algébriques n'est peut-être pas si différente qu'on le pensait de l'inférence de langages réguliers. Dans les langages réguliers, les structures se déduisent directement des chaînes. Quand, en revanche, il s'agit d'apprendre un langage algébrique à partir de chaînes, les structures possibles sont sous-déterminées. D'où l'idée, finalement assez naturelle, de chercher une forme normale pour représenter les langages algébriques qui contraigne le plus possible les structures sous-jacentes. Cette première approche demande bien sûr à être complétée et approfondie, notamment pour voir si elle recoupe ou non des travaux déjà effectués en inférence de grammaires algébriques.

Références

- ANGLUIN D. (1982). Inference of reversible languages. *J. ACM*, **29**(3), 741–765.
- BAR-HILLEL Y., GAIFMAN C. & SHAMIR E. (1960). On categorial and phrase structure grammars. *Bulletin of the Research Council of Israel*, **9F**.
- BESOMBES J. & MARION J.-Y. (2004). Learning reversible categorial grammars from structures. In *Categorial Grammars*.
- BUSZKOWSKI W. & PENN G. (1990). Categorial grammars determined from linguistic data by unification. *Studia Logica*, **49**, 431–454.
- DENIS F., LEMAY A. & TERLUTTE A. (2002). Some language classes identifiable in the limit from positive data. In *ICGI 2002*, number 2484 in *Lecture Notes in Artificial Intelligence*, p. 63–76 : Springer Verlag.
- DUPONT P., MICLET L. & VIDAL E. (1994). What is the search space of the regular inference. In L. N. IN ARTIFICIAL INTELLIGENCE, Ed., *ICGI'94 - Lectures Notes in Computer Science*, volume 862 - Grammatical Inference and Applications, p. 25–37, Heidelberg.
- FLORÊNCIO C. C. (2002). Consistent identification in the limit of rigid grammars from strings is np-hard. In M. V. Z. P. ADRIAANS, H. FERNAU, Ed., *Grammatical Inference : Algorithms and Applications*, volume 2484 of *Lecture Notes in Artificial Intelligence*, p. 49–62 : Springer Verlag.
- FLORÊNCIO C. C. (2001). Consistent identification in the limit of any of the classes k -valued is NP-hard. In *Logical Aspects of Computational Linguistics*, volume 2099 of *Lecture Notes in Artificial Intelligence*, p. 125–134 : Springer Verlag.
- GOLD E. (1967). Language identification in the limit. *Inform. Control*, **10**, 447–474.
- JOSHI A. & SCHABES Y. (1997). *Handbook of Formal Languages, vol3*, chapter Tree-Adjoining Grammars, p. 69–120. Springer Verlag.
- KANAZAWA M. (1998). *Learnable Classes of Categorial Grammars*. The European Association for Logic, Language and Information. CLSI Publications.
- ONCINA J. & GARCIA P. (1992). Inferring regular languages in polynomial update time. In *Pattern Recognition and Image Analysis*, p. 49–61.
- SAKAKIBARA Y. (1992). Efficient learning of context-free grammars from positive structural examples. *Information and Computation*, **97**(1), 23–60.
- WOODS W. (1970). Transition network grammars for natural language analysis. *Commun. ACM*, **10**, 591–606.