Apprentissage des grammaires catégorielles à partir de structures

Jérôme Besombes[1], Jean-Yves Marion[2]

[1] LORIA
615, rue du jardin botanique
54602 Villers-lès-Nancy, France
[2] INPL-Ecole Nationale Supérieure des Mines de Nancy
Parc de Saurupt,
54042 Nancy CEDEX, France
{Jerome.Besombes, Jean-Yves.Marion}@loria.fr

Résumé : Nous présentons $Alfa$, un algorithme d'apprentissage des grammaires catégorielles. Nous nous interessons à l'apprentissage à la limite à partir d'exemples positifs et les exemples sont des arbres de dérivations partiellement désetiquetés. Nous montrons qu'$Alfa$ identifie la classe des grammaires catégorielles réversibles, une classe contenant strictement les gramaires rigides. En ce sens, ce résultat constitue une extension de l'algorithme de Kanazawa d'apprentissage des grammaires rigides.

L'identification à la limite de Gold (Gold, 1967) définit un paradigme d'aprentissage particulèrement adapté à la compréhension et à la formalisation de l'acquisition du langage ; en effet, les travaux de Pinker (Pinker, 1994) ont conforté l'idée selon laquelle l'aprentissage s'effectue à partir de l'analyse de phrases correctes (des exemples positifs). D'autre part, Pinker émet l'hypothèse d'une structuration des phrases entendues (transformation du signal linéaire en arbre), cette structuration permettant la prise en compte d'informations syntaxiques et sémantiques utiles à la construction de grammaires et donc à l'apprentissage. Dans l'article (Besombes & Marion, 2002), nous avons défini un algorithme d'apprentissage des langages réguliers d'arbres réversibles à partir d'exemples positifs et avons montré que cet algorithme pouvait s'appliquer à des structures particulières : les arbres de dérivation. Dans cet article, nous allons étudier l'apprentissage de structures obtenues à partir d'arbres de dérivation de grammaires catégorielles. Largement utilisées pour la linguistique, ces grammaire basées sur des opérateurs logiques associent un vocabulaire et des ensembles de types qui représentent des catégories lexicales ; toute l'information de la grammaire est donc définie dans un lexique et l'on parle alors de grammaires lexicalisées. Les arbres de dérivation partiellement désetiquetés contiennent un information d'ordre sémantique. Par rapport à l'approche de Dudau-Sofronie dans (Dudau-Sofronie *et al.*, 2003), les informations sémantiques considérées dans cet article offrent un contexte d'apprentissage moins riche puisqu'il suffit de préciser des liens entre les mots. En outre, ce travail intègre l'idée

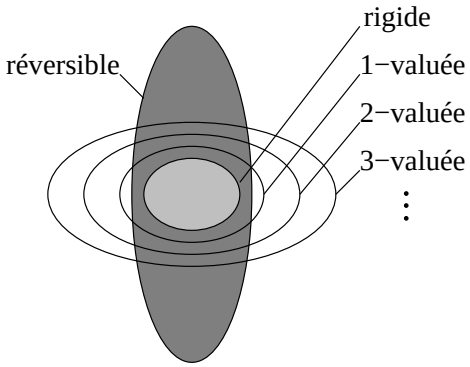


FIG. 1 – La classe des grammaires réversibles et le hiérarchie des k-valuées

selon laquelle le processus d'acquisition du langage a comme point de départ la notion de grammaire unverselle de Chomsky (Chomsky, 1986) et utilise des informations sémantiques pré-acquises. Dans (Kanazawa, 1998), Kanazawa a construit un algorithme d'apprentissage polynômial pour le cas des grammaires pour lesquelles un type unique est associé à chaque mot du vocabulaire (grammaires rigides). Plus généralement, pour le cas où k types au plus peuvent être associés à chaque mot du vocabulaire (grammaires k-valuées et rigides pour $k = 1$), il a été montré que si $k > 1$, l'apprentissage est un problème NP-complet (Florêncio, 2001). Pour prendre en compte les ambiguïtés des langues naturelles, la rigidité s'avère être une restriction particulièrement contraignante ; d'un autre côté, les grammaires k-valuées, non apprenables efficacement, dépendent d'une constante entière (une constante par grammaire), ce qui ne semble pas correspondre à une entité réaliste et mesurable. Suivant l'idée d'Angluin (Angluin, 1982), nous définissons une classe originale de grammaires catégorielles complétement indépendante de la hiérarchie des grammaires k-valuées (FIG. 1), contenant strictemment les rigides et apprenables en temps polynômial. Nous donnons un algorithme efficace d'apprentissage de cette classe et montrons sur des exemples, comment sont prises en compte des ambiguïtés linguistiques classiques.

1 Les grammaires catégorielles

1.1 Les grammaires catégorielles classiques

Nous définissons les grammaires catégorielles introduites dans (Y. Bar-Hillel & Shamir, 1960). Une grammaire catégorielle classique G est une relation entre deux ensembles Σ et Tp où :

- Σ est un vocabulaire fini,
- Tp est un ensemble de types défini à partir d'un ensemble fini de types primitifs Var comme le plus petit ensemble vérifiant :
  - Tp contient un type spécial s qui n'est pas un élément de Var (on note Pr

l'ensemble $Pr = Var \cup \{s\}$ des types primitifs),
- si $a \in Pr$ alors $a \in Tp$,
- si $A \in Tp$ et $B \in Tp$, alors $A\backslash B \in Tp$,
- si $A \in Tp$ et $B \in Tp$, alors $A/B \in Tp$.

On note $\alpha \mapsto_G A$ si $(\alpha, A) \in G$ et lorsqu'il n'y a pas d'ambiguité, on note simplement $\alpha \mapsto A$. Pour tout entier k, une grammaire est dite *k-valuée* si pour tout mot $\alpha \in \Sigma$, il existe au plus k distincts types $A_1, \ldots, A_k$ tels que $\forall 1 \leq i \leq k,\ \alpha \mapsto A_i$. Une grammaire 1-valuée est également dite *rigide*. Un élément A de Tp est un *sous-type* d'un élément C de Tp si et seulement si
- $C = A$
- ou $C = B\backslash B'$ et A est un sous-type de B ou de B'
- ou $C = B/B'$ et A est un sous-type de B ou de B'

Si α est un élément de Σ, l'ensemble fini $Cat_G(\alpha)$ est l'ensemble des types associés à α :

$$Cat_G(\alpha) = \{A \in Tp,\ G : \alpha \mapsto A\}.$$

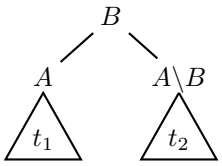
Un type a est dit *sous-type primitif* d'un type C si a est un sous-type de C et a est un type primitif. Un type A est dit *sous-type argument* d'un type C s'il existe un type B tel que :
- soit $A\backslash B$ est un sous-type de C
- soit B/A est un sous-type de C.

Un type A est dit *sous-type foncteur* d'un type C s'il existe un type B tel que :
- soit A/B est un sous-type de C
- soit $B\backslash A$ est un sous-type de C.

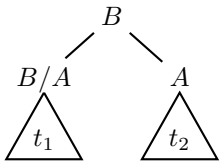
Un *type-contexte* $A[\sharp]$ est un type pour lequel un sous-type exactement a été remplacé par le symbole $\sharp$ n'appartenant pas à Tp. Pour tout type B, on note $A[B]$ le type obtenu par substitution de $\sharp$ par B dans $A[B]$.

Pour toute grammaire catégorielle G, un *arbre de dérivation partiel* pour G est un arbre binaire ordonné ayant une des formes suivantes :

pour tout $\alpha \in \Sigma$ et tout $A \in Tp$ tels que $G : \alpha \mapsto A$



pour tout arbre de dérivation partiel t_1 (appelé sous-arbre foncteur) de racine $> B/A$ et tout arbre de dérivation partiel t_2 (appelé sous-arbre argument) de racine A.



pour tout arbre de dérivation partiel t_1 (appelé sous-arbre argument) de racine A et tout arbre de dérivation partiel t_2 (appelé sous-arbre foncteur) de racine $A\backslash B$,

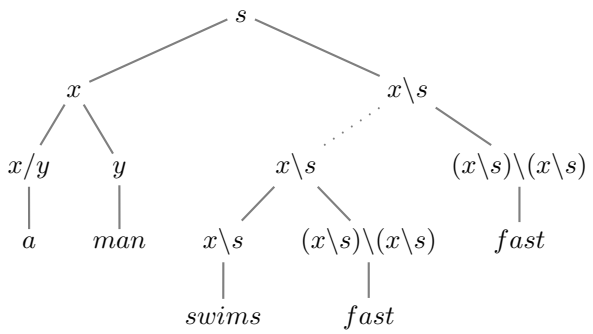
Un *arbre de dérivation* est un arbre de dérivation partiel dont la racine est s. L'ensemble des arbres de dérivation d'une grammaire G se note $PL(G)$. Nous avons vu qu'un tel ensemble $PL(G)$ est un langage régulier d'arbres. Le langage produit par une grammaire G est l'ensemble $\mathcal{L}_G \subset \Sigma^*$ des éléments du type $\alpha_1 \dots \alpha_n$ correspondant aux feuilles des arbres de dérivation lues de gauche à droite.

Une *FA-structure* (resp. *FA-structure partielle*) est un arbre $Structure(t)$ obtenu à partir d'un arbre de dérivation (resp. arbre de dérivation partiel) t en remplaçant l'étiquette de chaque nœud interne par $/$ si son fils gauche est un sous-arbre foncteur et son fils droit un sous-arbre argument, par $\backslash$ si son fils gauche est un sous-arbre argument et son fils droit un sous-arbre foncteur, et par l'étiquette de son fils si celui-ci est unique (ce qui, par définition des arbres de dérivation implique que ce fils est une feuille). L'ensemble des FA-structures d'un grammaire G est noté $FL(G)$. Si s est une FA-structure, on note $Cat_G(s)$ l'ensemble des types A tels qu'il existe un arbre de dérivation partiel t tel que $Structure(t) = s$ et tel que la racine de t est étiquetée par A. Un *FA-contexte* est un contexte obtenu en remplaçant un sous-arbre d'une FA-structure par $\sharp$.

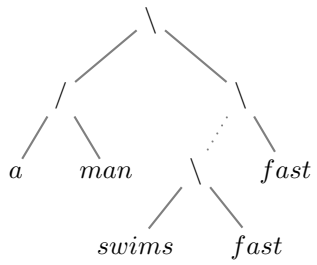
Exemple 1
Soit G la grammaire catégorielle défi nie par :

$$G : \quad \begin{array}{rcc} a & \mapsto & x/y \\ fast & \mapsto & (x\backslash s)\backslash(x\backslash s) \\ man & \mapsto & y \\ swims & \mapsto & x\backslash s \end{array}$$

Les arbres de dérivation de G sont les arbres de la forme :



Les FA-structures de G sont les arbres de la forme :



Le langage produit par G est donc :

$$\mathcal{L}_G = \{a\ man\ swims\ \underbrace{fast \ldots fast}_{n \text{ fois, } n \geq 0}\}$$

Un *substitution* σ est une fonction de Var dans Tp. Une substitution est étendue à une fonction de Tp dans Tp par :

$$\sigma(A\backslash B) = \sigma(A)\backslash\sigma(B)$$

et

$$\sigma(B/A) = \sigma(B)/\sigma(A)$$

et définit une grammaire $G' = \sigma(G)$:

$$G' : \alpha \mapsto A' \Leftrightarrow \exists A \in Var,\ A' = \sigma(A) \text{ et } G : \alpha \mapsto A$$

Lemme 1 (Buszkowski et Penn (Buszkowski & Penn, 1990))
Si G est une grammaire et σ une substitution, alors $FL(G) \subseteq FL(\sigma(G))$.

Une *projection* est une substitution de Var dans Var et un *renommage* est une projection bijective.

1.2 Les grammaires catégorielles compactes

Un type est dit *compact* si et seulement si il a une des formes suivantes :
– $C = a$, où a est un type primitif,
– $C = B/a$, où a est un type primitif et où B est compact,
– $C = a\backslash B$, où a est un type primitif et où B est compact.

En d'autres termes, un type est compact si et seulement si il est d'ordre au plus 1, où l'ordre o d'un type quelconque est défini par :
– $(a) = 0$ pour tout type a primitif,
– $o(B/C) = o(C\backslash B) = max\{o(B), o(C) + 1\}$

Une grammaire G est *compacte* si tout élément de $Cat(G) = \{Cat_G(\alpha)\ :\ \alpha \in \Sigma\}$ est compact.

Théorème 1
Pour toute grammaire catégorielle G, il existe une grammaire catégorielle compacte G' telle que $FL(G) = FL(G')$.

Démonstration :[Idée] L'algorithme de la figure 2 permet de transformer une grammaire catégorielle quelconque G en une grammaire catégorielle compacte $G' = Compact(G)$ telle que $FL(G) = FL(G')$.

□

Entrée : une grammaire catégorielle G
Sortie : une grammaire catégorielle compacte $Compact(G)$
telle que $LG(G) = LG(G')$
Initialisation : $G' = G$
DEBUT
TANT QUE G' n'est pas compacte FAIRE
Choisir un type C dans $Cat(G')$ avec un sous-type argument A non primitif non primitif
Choisir un nouveau type primitif $a \notin Var$
Pour tout type B dans $Cat(G')$
SI A est un sous-type argument de B
ALORS pour tout $\alpha \mapsto B$ dans G'
remplacer A par a dans B
Pour tout type B dans $Cat(G')$
SI A est un sous-type foncteur de B,
ALORS pour tout $\alpha \mapsto B$ dans G' ajouter $\alpha \mapsto B'$ dans G',
avec B' le type obtenu en remplaçant A par a dans B
(G' conserve $\alpha \mapsto B$)
FIN SI
FIN TANT QUE
Retourner G' en sortie
FIN

FIG. 2 – Algorithme de calcul de $Compact(G)$

Exemple 2
La grammaire G définie dans l'exemple 1 n'est pas compacte puisque $(x\backslash s)$ est un sous-type argument de $(x\backslash s)\backslash(x\backslash s)$. La grammaire compacte correspondante G' est définie par :

$$G' : \begin{array}{rcll} a & \mapsto & x/y & \\ man & \mapsto & y & \\ swims & \mapsto & z, & x\backslash s \\ fast & \mapsto & z\backslash(x\backslash s), & z\backslash z \end{array}$$

où z est le nouveau type primitif introduit pendant l'exécution de l'algorithme.

1.3 Les grammaires catégorielles réversibles

Un ensemble de types $\Gamma \subseteq Tp$ est dit *réversible* s'il n'existe pas deux types dans Γ qui ne diffèrent que d'un type primitif unique. Formellement, un ensemble de types $\Gamma \subseteq Tp$ est dit *réversible* si pour tout type-contexte $A[\sharp]$, il existe au plus un type primitif a tel que $A[a]$ appartient à Γ.

Une grammaire catégorielle G est *réversible* si G est compacte et si pour tout $\alpha \in \Sigma$, l'ensemble de types $Cat_G(\alpha)$ est réversible.

Exemple 3
La grammaire G' de l'exemple 2 est réversible.
La grammaire G'' défi nie par :

$$G'' : \begin{array}{rcl} a & \mapsto & x/y \\ man & \mapsto & y \\ swims & \mapsto & z_1, \quad x\backslash s \\ fast & \mapsto & z_1\backslash(x\backslash s), \quad z_2\backslash(x\backslash s), \quad z_1\backslash z_2 \end{array}$$

n'est pas réversible. En effet, $fast \mapsto A[z_1]$ et $fast \mapsto A[z_2]$, où $A[\sharp]$ est le type-contexte $\sharp\backslash(x\backslash s)$. On notera que l'on a :

$$\mathcal{L}_{G''} = \{a\ man\ swims,\ a\ man\ swims\ fast,\ a\ man\ swims\ fast\ fast\}$$

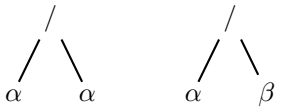
Montrons que les grammaires catégorielles réversibles sortent du cadre des langages d'arbres réversibles.

Théorème 2
Il existe des grammaires catégorielles réversibles G dont les ensembles de FA-structures $FL(G)$ ne sont pas des langages réguliers d'arbres réversibles.

Démonstration : Considérons la grammaire réversible G définie par :

$$G : \begin{array}{rcl} \alpha & \mapsto & s/a \\ \alpha & \mapsto & a \\ \beta & \mapsto & a \end{array}$$

Alors l'ensemble des FA-structures produites par G est le couple :



Ce couple est un langage régulier d'arbres dont l'automate minimal $\mathcal{A}_G$ est :

$$\begin{array}{rcl} a(q_1, q_1) & \xrightarrow[\mathcal{A}_G]{} & q_0 \\ a(q_1, q_2) & \xrightarrow[\mathcal{A}_G]{} & q_0 \\ \alpha & \xrightarrow[\mathcal{A}_G]{} & q_1 \\ \beta & \xrightarrow[\mathcal{A}_G]{} & q_2 \end{array}$$

où q_0 est l'unique état final. $\mathcal{A}_G$ n'est pas un automate réversible.

Théorème 3
Pour toute grammaire catégorielle rigide G il existe une grammaire réversible G' telle que $FL(G) = FL(G')$.

Démonstration : soit G une grammaire rigide et montrons que $Compact(G)$ est réversible. L'algorithme de calcul de $Compact(G)$ modifie G de deux manières : il remplace des sous-types A par des sous-types primitifs a et il ajoute des nouvelles relations $\alpha \mapsto B'$. Puisque la grammaire G est rigide, pour tout élément α de Σ, l'ensemble $Cat_G(\alpha)$ est un singleton et est donc réversible. Il est maintenant facile de vérifier que les deux types d'actions décrites ci-avant conservent la réversibilité de ces ensembles $Cat_G(\alpha)$. Puisqu'enfin $Compact(G)$ est compacte, alors elle est réversible, ce qui, d'après le théorème 1 permet de conclure.

□

Théorème 4
Soient G une grammaire catégorielle réversible et u une FA-structure de $FL(G)$. Il existe un unique arbre de dérivation t dans $PL(G)$ tel que $u = structure(t)$.

Nous allons démontrer ce théorème à l'aide du lemme suivant.

Lemme 2
Soient G une grammaire catégorielle réversible et u une FA-structure de $FL(G)$ et $A[\sharp]$ un type-contexte. Il existe alors au plus un unique type primitif a tel que le type $A[a]$ soit dans $Cat_G(u)$.

Démonstration :[Idée] par induction sur la taille de u.

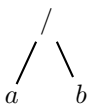
Démonstration du théorème 4 : d'après le lemme 2, si G est une grammaire réversible, alors Cat_G est une fonction de l'ensemble des sous-arbres arguments de $FL(G)$ dans l'ensemble des types primitifs Tp. Cette fonction induit alors une bijection de $FL(G)$ dans $PL(G)$ correspondant à l'inverse de la fonction $Structure$. □

Remarque 1
Le théorème 4 établit qu'à une FA-structure correspond au plus un unique arbre de dérivation. Ce résultat n'est pas vrai pour les arbres de dérivation partiels. Considérons l'exemple suivant.

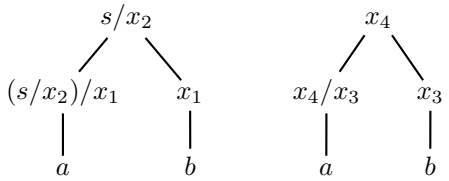
Exemple 4
Soit $G^{(3)}$ la grammaire définie par :

$$G^{(3)} : \begin{array}{lcll} a & \mapsto & (s/x_2)/x_1, & x_4/x_1 \\ b & \mapsto & x_1 & \\ c & \mapsto & x_2, & s/x_4 \end{array}$$

A la FA-structure partielle :



correspondent les deux arbres de dérivation partiels suivants :



La propriété d'unicité d'arbre de dérivation pour une FA-structure donnée est donc plus faible que dans le cas des grammaires rigides pour lesquelles la fonction $Structure$ est bijective pour les arbres de dérivation partiels.

Théorème 5
Soient G une grammaire réversible et $\mathtt{f}[\sharp]$ un FA-contexte pour G. Il existe un unique type A tel que pour une FA-structure u, $\mathtt{f}(u)$ soit dans $FL(G)$ et $Cat_G(u) = A$.

Démonstration :[Idée] l'existence d'un tel type A est une conséquence directe de la définition d'une FA-structure. Nous montrons ensuite que A est unique par induction sur la profondeur du symbole $\sharp$ dans le contexte $\mathtt{f}[\sharp]$.

On note $Cat_G(f[\sharp])$ le type A défini comme pour le théorème 5. On pourra voir Cat_G comme une fonction de l'ensemble des FA-contextes dans Tp. Nous retrouvons une notion de déterminisme descendant (unicité d'un état ou d'un type correspondant à un contexte donné). Dans le cas des langages réguliers d'arbres, cette propriété n'est pas suffisante à l'apprentissage ; nous allons voir que dans le cas particulier des FA-structures, la propriété de déterminisme descendant suffit pour conclure en l'apprenabilité de la classe des grammaires catégorielles réversibles.

1.4 Ensemble caractéristique

Comme pour le cas des langages d'arbres, nous allons définir pour toute grammaire réversible G, un sous ensemble fini de $FL(G)$ qui pourra nous garantir l'apprentissage à la limite.

Nous associons à tout type primitif a de G un FA-contexte $Leaf_G(a)$ tel que :

$$Cat_G(Leaf_G(a)) = a$$

et une FA-structure $Root_G(a)$ telle que :

$$a \in Cat_G(Root_G(a)).$$

Pour le cas spécial du type s, on choisit

$$Leaf_G(s) = \sharp.$$

A tout type non-primitif A, on associe un FA-structure $Root_G(A)$ telle que

$$A \in Cat_G(Root_G(A))$$

et si A appartient à $Cat(G)$ ($\alpha \mapsto A$ pour un élément α de Σ), on choisit

$$Cat_G(Root_G(A)) = \alpha.$$

Enfin, pour chaque type non primitif A, on définit un FA-contexte défini de manière récursive par :

– si $A = B/a$, $Leaf_G(A) = Leaf_G(B)\left[\begin{array}{c} / \\ \diagup \quad \diagdown \\ \sharp \qquad Root_G(a) \end{array}\right]$

– si $A = a\backslash B$, $Leaf_G(A) = Leaf_G(B)\left[\begin{array}{c} \backslash \\ \diagup \quad \diagdown \\ Root_G(a) \qquad \sharp \end{array}\right]$

Un ensemble fini de FA-structures est un *ensemble caractéristique* pour une grammaire réversible G s'il contient pour tout type A de G la structure suivante :

$$Leaf_G(A)[Root_G(A)].$$

Un tel ensemble est noté $\mathcal{C}(G)$ et l'on vérifie aisément que $\mathcal{C}(G) \subseteq FL(G)$.

Nous allons maintenant voir que nous retrouvons le résultat classique pour les ensembles caractéristiques, résultat qui va nous permettre de conclure immédiatement à l'apprenabilité des grammaires catégorielles réversibles à partir de FA-structures.

Lemme 3
Si G et G' sont deux grammaires catégorielles réversibles et $\mathcal{C}(G)$ un ensemble caractéristique pour G tel que $\mathcal{C}(G) \subseteq FL(G')$ alors $FL(G) \subseteq FL(G')$.

Démonstration : pour démontrer cette inclusion, nous allons définir une projection σ telle que $\sigma(G) \subseteq G'$, ce qui, d'après le lemme 1 permettra de conclure. D'après la définition d'un ensemble caractéristique, pour tout type primitif a de G, il existe une FA-structure $Leaf_G(a)[Root_G(a)]$ dans $\mathcal{C}(G)$. Puisque $CS(G) \subset FL(G')$, le FA-contexte $Leaf_G(a)[\sharp]$ est également un FA-contexte pour G'. De plus, $root_G(a)$ est un sous-arbre argument de $Leaf_G(a)[\sharp]$ et G' est réversible, ce qui implique que $Cat_{G'}(Leaf_G(a)[\sharp])$ est un type primitif de G'. On définit alors $\sigma(a) = a'$ et σ est classiquement étendue aux types non-primitifs par $\sigma(A\backslash b) = \sigma(A)\backslash\sigma(b)$ et $\sigma(b/A) = \sigma(b)/\sigma(A)$. Montrons maintenant par induction que pour tout type A de G, $Cat_{G'}(Leaf_G(A)) = \sigma(A)$.

– si A est un type primitif, le résultat est donné par le définition de σ,

– si $A = B\backslash a$. Par définition, le FA-contexte $Leaf_G(B\backslash a)$ est de la forme

$$Leaf_G(A) = Leaf_G(B)\left[\begin{array}{c} \backslash \\ \diagup \quad \diagdown \\ Root_G(a) \qquad \sharp \end{array}\right]$$

Par hypothèse d'induction, $Cat_{G'}(Leaf_G(B)) = \sigma(B)$. De plus, $Leaf_G(a)[Root_G(a)]$ est dans $CS(G) \subseteq FL(G')$, ce qui implique que $\sigma(a)$ est un type primitif dans $Cat_{G'}(Root_G(a))$ et donc, d'après le lemme 2, $\sigma(a)$ est un type primitif unique de $Cat_{G'}(Root_G(a))$. On a donc $Cat_{G'}(Leaf_G(B\backslash a))$ qui est égale à $\sigma(B)\backslash\sigma(a) = \sigma(B\backslash a)$.

– Si $A = a/B$. Ce cas peut être traité symétriquement au cas précédent. Considérons $\alpha \mapsto_G A$. Par définition, on a $Root_G(A) = \alpha$ et $Cat_{G'}(Leaf_G(A)) = \sigma(A)$ ce qui implique que $\alpha \mapsto_{G'} \sigma(A)$ et finalement que $\sigma(G) \subseteq G'$.

□

2 L'algorithme $Alfa$

Suivant (Gold, 1967), nous définissons l'identification à la limite à partir d'exemples positifs. Uspensky et Shen (Uspensky & Shen, 1996) introduisent la notion de mode de description que nous allons utiliser pour la définition du paradigme d'apprentissage. Soient Σ et Π deux alphabets ; un mode de descripion d est une relation récursivement énumerable dans $\Sigma^* \times \Pi^*$. Un code c de Π^* définit un langage $L = L(c)$ par $L(c) = \{\omega \in \Sigma^* \ : \ (\omega, c) \in d\}$. Une classe de langage $\mathbb{L}$ est un ensemble $\mathbb{L} = \{L(c) \ : \ c \in \Pi^*\}$.

Une *présentation positive* s d'un langage L est une surjection de $\mathbb{N}$ dans L. On note alors $s[n]$ la séquence finie $s(0), \ldots, s(n)$ issue de s . L'ensemble des séquences finies issues des présentations positives des langages d'une classe $\mathbb{L}$ est noté $S(\mathbb{L})$. Une fonction I de $S(\mathbb{L}) \mapsto \Pi^*$ converge vers un langage L si pour toute présentation s de L, il existe un entier N tel que pour tout $n > N$ on a $I(s[n]) = c$ et $L(c) = L$. La fonction I identifie la classe $\mathbb{L}$ si I converge vers L pour tout langage L et pour toute présentation positive s de L.

Dans notre cas, les codes sont des grammaires. Une classe de grammaires categorielles Π est dite *apprenable à partir de structures* si l'ensemble $FL(\Pi) = \{FL(G) \ : \ G \in \Pi\}$ est identifiable.

Théorème 6
La classe des grammaires catégorielles réversibles est apprenable à partir de FA-structures.

Démonstration : on vérifie facilement que les ensembles caractéristiques possèdent toutes les propriétés des ensembles telltale, le résultat d'Angluin (Angluin, 1982) permet de conclure. □

Nous allons maintenant définir un algorithme d'apprentissage efficace .

Théorème 7
L'algorithme $Alfa$ apprend la classe des grammaires catégorielles réversibles à partir de FA-structures.

Démonstration : soit $\mathtt{f}_1, \mathtt{f}_2, \ldots$ une énumération de l'ensemble $FL(G)$ des FA-structures d'une grammaire catégorielle réversible G. Pour tout entier p, l'exécution d'$Alfa$ sur l'entrée $\mathtt{f}_1, \ldots, \mathtt{f}_p$ construit une série de grammaires $G_0, \ldots, G_n$. Puisque pour tout i, $|Cat(G_{i+1}| < |Cat(G_i)|$ et puisque $Cat(G_0)$ est fini, l'algorithme s'arrête après n étapes et, par définition, la grammaire G_n est réversible. La première étape qui correspond au calcul de G_0, est similaire à celui de l'algoritme RG (Kanazawa, 1998) d'apprentissage des grammaires rigides. G_0 vérifie $FL(G_0) = \{\mathtt{f}_1, \ldots, \mathtt{f}_p\}$ et il existe un substitution θ telle que $\theta(G_0) = G$. Puisque G est réversible, pour tous types primitifs a_1 et a_2, si $\sigma_{a_1 \leftarrow a_2}$ est la substitution appliquée pour construire G_{i+1} à partir de G_i,

Entrée : un ensemble fini de FA-structures E
Sortie : une grammaire réversible G
DEBUT
  Construction d'une grammaire G_0 en introduisant un nouveau type primitif
  pour chaque sous-arbre argument de E et en complétant ensuite les FA-structures
  pour obtenir des arbres de dérivation (les racines sont étiquetées par s)
  TANT QUE G_i n'est par réversible FAIRE
    trouver un élément α tel que $\alpha \mapsto_G A[a_1]$ et $\alpha \mapsto_G A[a_2]$,
    avec a_1 et a_2 deux type primitifs
    Construire $G_{i+1} = \sigma_{a_1 \leftarrow a_2}(G_i)$,
    où $\sigma_{a_1 \leftarrow a_2}$ est la projection définie par $\sigma_{a_1 \leftarrow a_2}(a_1) = a_2$
    et $\sigma_{a_1 \leftarrow a_2}(a) = a$ pour tout a différent de a_1
  FIN TANT QUE
  Retourner G_n en sortie
FIN

FIG. 3 – L'algorithme $Alfa$

pour i entre 0 et $n-1$, alors $\theta(a_1) = \theta(a_2)$, ce qui montre qu'il existe une substitution σ' vérifiant $\theta = \sigma' \circ \sigma$. Alors d'après le lemme 1, on a :

$$FL(G_0) \subseteq FL(\sigma(G_0)) \subseteq FL(\sigma'(\sigma(G_0)))$$

et donc

$$FL(G_n) \subseteq FL(G).$$

Enfin, puisque $FL(G_0) = \{\mathtt{f}_1, \ldots, \mathtt{f}_p\}$ et puisque $FL(G_0) \subseteq FL(G_n)$, pour n assez grand, l'ensemble donné en entrée contient un ensemble caractéristique pour G, ce qui, d'après le lemme 3 donne

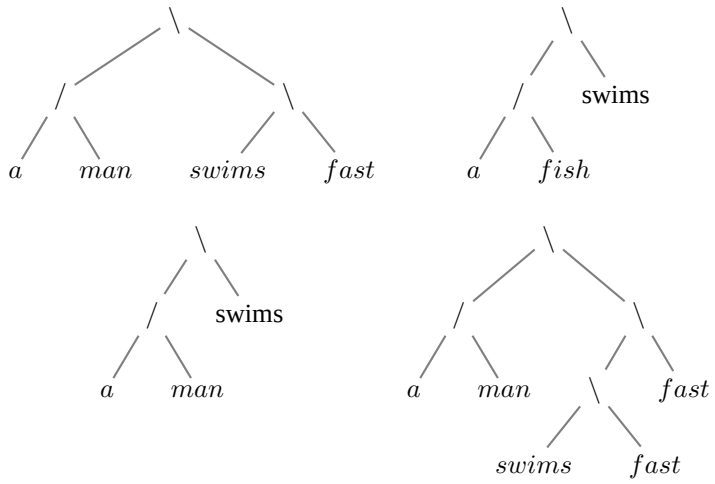
$$FL(G) \subseteq FL(G_n)$$

et permet de conclure.

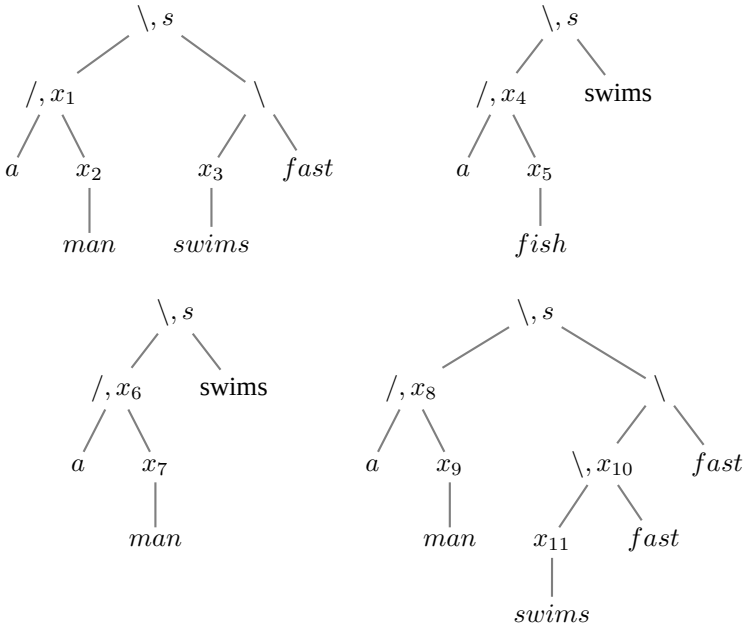
L'algorithme fonctionne en temps polynomial en fonction de la taille n des entrées, la construction de G_0 nécessitant un temps linéaire, le nombre d'étapes total étant inférieur ou égal à n et la vérification de la réversibilité de chaque G_i étant polynomial en fonction de $|Cat(G_i)|$ qui est également inférieur à n. Nous allons maintenant illustrer $Alfa$ sur des exemples.

3 Exemples

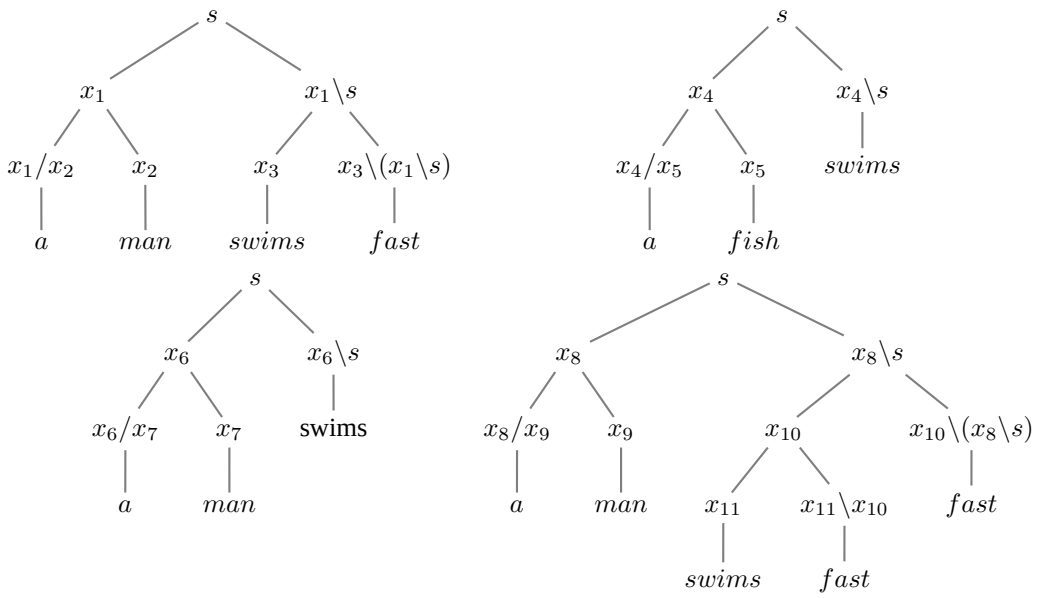
Le premier exemple est celui de l'apprentissage de la grammaire donnée en exemple par Kanazawa dans (Kanazawa, 1998). Considérons donc l'entrée définie par les quatre FA-strutures suivantes.



$Alfa$ commence par étiqueter les arbres en commençant par les racines des sous-arbres arguments ; un nouveau type primitif étant introduit pour chaque sous-arbre ; les racines des arbres d'entrée sont étiquetées par s.



Les arbres sont alors complétés en arbres de dérivation.



La première grammaire G_0 est alors construite. Elle est définie par :

$$
\begin{array}{rrlllll}
 & a & \mapsto & x_1/x_2, & x_4/x_5, & x_6/x_7, & x_8/x_9 \\
 & man & \mapsto & \mathbf{x_2}, & \mathbf{x_7}, & \mathbf{x_9} & \\
G_0: & fish & \mapsto & x_5, & & & \\
 & swims & \mapsto & \mathbf{x_3}, & \mathbf{x_4}\backslash s, & \mathbf{x_6}\backslash s, & \mathbf{x_{11}} \\
 & fast & \mapsto & x_3\backslash(x_1\backslash s), & x_{11}\backslash x_{10}, & x_{10}\backslash(x_8\backslash s) &
\end{array}
$$

Les substitutions suivantes sont alors effectuées :

$$x_2 = x_7 = x_9$$

$$x_3 = x_{11}$$

$$x_4 = x_6$$

ce qui permet d'obtenir une grammaire G_1 définie par :

$$
\begin{array}{rrlllll}
 & a & \mapsto & \mathbf{x_1}/x_2, & x_4/\mathbf{x_5}, & \mathbf{x_4}/\mathbf{x_2}, & \mathbf{x_8}/x_2, \\
 & man & \mapsto & x_2 & & & \\
G_1: & fish & \mapsto & x_5 & & & \\
 & swims & \mapsto & x_3, & x_4\backslash s & & \\
 & fast & \mapsto & x_3\backslash(x_1\backslash s), & x_3\backslash x_{10}, & x_{10}\backslash(x_8\backslash s) &
\end{array}
$$

Ce sont alors les substitutions suivantes qui sont appliquées :

$$x_2 = x_5$$

$$x_1 = x_4 = x_8$$

ce qui donne la grammaire G_2 :

$$\begin{array}{rcll} & a & \mapsto & x_1/x_2 \\ & man & \mapsto & x_2 \\ G_2 : & fish & \mapsto & x_2 \\ & swims & \mapsto & x_3, \quad x_1\backslash s \\ & fast & \mapsto & \mathbf{x_3}\backslash(x_1\backslash s), \quad x_3\backslash x_{10}, \quad \mathbf{x_{10}}\backslash(x_1\backslash s) \end{array}$$

Enfin, l'ultime substitution

$$x_3 = x_{10}$$

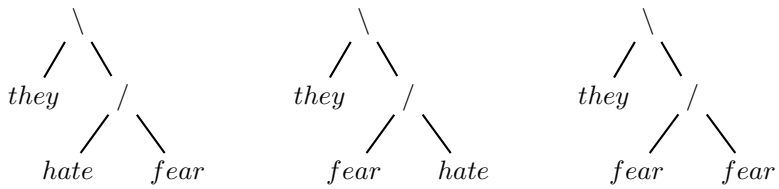
permet de calculer G_3 :

$$\begin{array}{rcll} & a & \mapsto & x_1/x_2 \\ & man & \mapsto & x_2 \\ G_3 : & fish & \mapsto & x_2 \\ & swims & \mapsto & x_3, \quad x_1\backslash s \\ & fast & \mapsto & x_3\backslash(x_1\backslash s), \quad x_3\backslash x_3 \end{array}$$

G_3 est réversible et l'algorithme s'arrête.

Remarque 2
L'exemple précédent, initialement donné par Kanazawa pour illustrer l'algorithme d'apprentissage des grammaires catégorielles rigides, montre que cet apprentissage ne se réalise pas de la même manière que pour notre algorithme d'apprentissage des grammaires catégorielles réversibles. En effet, si les arbres correspondant aux phrases « a man swims fast » et « a fish swims » suffit à Kanazawa pour retouver la grammaire, les quatre arbres donnés en entrée ici sont nécessaires à l'apprentissage. Il apparaît que la méthode de généralisation diffère pour les deux algorithmes.

Nous allons maintenant considérer des grammaires non-rigides qui illustrent des exemples linguistiques concrets. Sur l'entrée suivante :

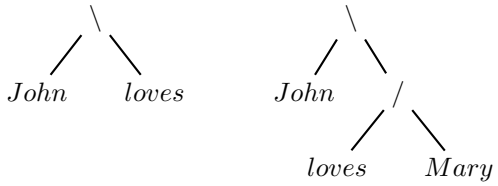


$Alfa$ calcule la grammaire suivante :

$$\begin{array}{rcll} & they & \mapsto & x \\ G : & hate & \mapsto & (x\backslash s)/y, \quad y \\ & fear & \mapsto & (x\backslash s)/y, \quad y \end{array}$$

Pour laquelle aucune grammaire rigide équivalente n'existe.

Enfin sur l'entrée suivante :



$Alfa$ calcule la grammaire

$$G: \begin{array}{lcll} John & \mapsto & x & \\ loves & \mapsto & x\backslash s, & (x\backslash s)/y \\ Mary & \mapsto & y & \end{array}$$

qui, encore une fois, n'a pas d'équivalent rigide.

Nous avons défini un algorithme efficace d'apprentissage de la classe des grammaires catégorielles réversibles à partir de FA-structures. Cette classe originale permet de capturer des ambiguïtés linguistiques que la trop grande restriction des grammaires catégorielles rigides ne prend pas en compte, et cela en garantissant la possibilité d'un apprentissage efficace. Il s'agit donc là d'un résultat très satisfaisant alliant l'expressivité d'un formalisme et l'apprenabilité de celui-ci.

Références

ANGLUIN D. (1982). Inference of reversible languages. *Journal of the ACM*, **29**, 741–765.

BESOMBES J. & MARION J. (2002). Apprentissage des langages réguliers d'arbres et applications. *Conférence d'Apprentissage, Orléans 17, 18 et 19 juin 2002*, p. 55–70.

BUSZKOWSKI W. & PENN G. (1990). Categorial grammars determined from linguistic data by unification.

CHOMSKY N. (1986). *Knowlege of Language*. Praeger, New York.

DUDAU-SOFRONIE D., TELLIER I. & TOMMASI M. (2003). Une classe de grammaires catégorielles apprenable à partir d'exemples typés. *Conférence d'Apprentissage, Laval du 1er au 4 juillet 2003*, p. 169–184.

FLORÊNCIO C. C. (2001). Consistent identification in the limit of any of the classes -valued is np-hard. In C. R. P. DE GROOTE, G. MORRILL, Ed., *Logical Aspects of Computational Linguistics*, Lecture Notes in Computer Science, p. 125–138 : Springer-Verlag.

GOLD M. (1967). Language identification in the limit. *Information and Control*, **10**, 447–474.

KANAZAWA M. (1998). *Learnable classes of Categorial Grammars*. CSLI.

PINKER S. (1994). *The language instinct*. Harper.

USPENSKY V. & SHEN A. (1996). Relations between varietes of kolmogorov complexities. **29**, 271–292.

Y. BAR-HILLEL C. G. & SHAMIR E. (1960). On categorial and phrase structure grammars. *Bulletin of Research Council of Israel*, **F**(9), 1–16.