# APPRENTISSAGE DES LANGAGES RÉGULIERS D'ARBRES ET APPLICATIONS

Jérôme Besombes, Jean-Yves Marion*

**Résumé**

Nous nous intéressons à l'apprentissage des langages réguliers d'arbres à partir d'exemples positifs. Notre modèle d'apprentissage est l'identification à la limite de Gold et les exemples sont des arbres. Nous démontrons que les langages réguliers d'arbres réversibles sont identifiables. Nous présentons trois conséquences de ce résultat. (1) Nous montrons que les grammaires algébriques *reset-free* sont identifiables à partir des arbres de dérivation. (2) Nous montrons que les langages d'arbres de dépendances sont identifiables. (3) Nous donnons une nouvelle démonstration de l'apprentissage des grammaires catégorielles rigides.

## 1 INTRODUCTION

Des linguistes comme Chomsky et Pinker (Pinker, 1994), ont montr´e que l'acquisition des langages naturels est bas´ee sur l'analyse de structures de phrases correctes. Ceci illustre l'importance de l'identifi cation `a la limite `a partir d'exemples positifs de Gold. Notre objectif est de contribuer `a la formalisation et `a la compr´ehension du processus d'acquisition du langage. Dans ce but, nous nous basons sur les travaux d'Angluin et Sakakibara concernant l'apprentissage. Nous montrons que la classe des langages r´eguliers d'arbres r´eversibles est identifi able. Tous ces r´esultats issus de motivations diff´erentes, partagent un point de vue commun ; ils supposent qu'un langage est produit par un m´ecanisme fi ni et r´eversible et ainsi, qu'il doit être possible d'inverser une production. Angluin (Angluin, 1982) a montr´e que les langages de mots reconnus par des automates d'´etats fi nis r´eversibles sont identifi ables, puis, Sakakibara (Sakakibara, 1992) a propos´e une notion de grammaires alg´ebriques r´eversibles. Pour ce dernier, les donn´ees ne sont plus des mots mais des structures d'arbres correspondant `a des squelettes d'arbres de d´erivation. Ainsi, le probl`eme est r´eduit `a l'identifi cation d'une sous-classe de langages r´eguliers d'arbres r´eversibles et les exemples sont des ´el´ements du langages. Nous avons ´evolu´e de l'apprentissage de grammaires `a l'apprentissage de langages, avec l'avantage d'une plus grande simplicit´e conceptuelle. Notre r´esultat peut s'appliquer `a diff´erents types de grammaires :

1. les grammaires de d´ependances r´eversibles sont identifi ables `a partir d'exemples positifs. Voir le th´eor`eme 30,

* LORIA, BP 239. 54506 Vandoeuvre-l`es-Nancy

2. les grammaires cat´egorielles rigides sont identifiables `a partir des squelettes d'arbres de d´erivation. Voir le th´eor`eme 30 et la nouvelle d´emonstration de ce r´esultat ´etabli par Kanasawa (Kanazawa, 1998).

D'un point de vue pratique, il apparait que nous avons acc`es aux ´el´ements d'un langage, mais pas au m´ecanisme de production de ces ´el´ements. Les langages d'arbres r´eguliers repr´esentent un cadre formel g´en´eral pour des domaines aussi vari´es que les documents XML ou le langage naturel ce qui nous int´eresse particuli`erement. L'id´ee de d´epart ´etant que les donn´ees sont des arbres annot´es et que ces annotations repr´esentent des informations s´emantiques et/ou syntaxiques qui, une fois prises en compte, doivent faciliter l'apprentissage. De telles donn´ees se retrouvent dans un grand nombre de grammaires formelles de linguistique computationnelle. Mentionnons par exemple les LFG, les TAg ou encore les grammaires cat´egorielles.

***Travaux reliés***

Un certain nombre d'articles concernent l'apprentissage de grammaires d'arbres. Nous avons d´ej`a mentionn´e les travaux d'Angluin et de Sakakibara (Angluin, 1982; Sakakibara, 1992) ; d'autres (Edwards *et al.*, 1976; Levine, 1981; Kamata, 1988; Fukuda & Kamata, 1984) se basent sur l'id´ee d'inf´erence *k-tail* des langages r´eguliers de mots de Biermann et Feldman (Biermann & Feldman, 1972) puis Knuutila et Steinby (Knuutila & Steinby, 1994). R´ecemment, Fernau (Fernau, 2001) a utilis´e ce cadre pour les grammaires XML. Carrasco et Oncina dans (Edwards *et al.*, 1976) ont montr´e que les langages r´eguliers d'arbres sont identifiables `a partir d'exemples probabilistes. Le lecteur int´eress´e par l'apprentissage des langages alg´ebriques peut consulter les ´etudes (Lee, 1996; Sakakibara, 1997; Mäkinen, 1997).

# 2 LANGAGES RÉGULIERS D'ARBRES

## 2.1 Les arbres comme des termes

Le livre (Comon *et al.*, 1997) donne les bases de la th´eorie des langages r´eguliers d'arbres. Un alphabet ordonn´e est un couple $(\mathcal{F}, arite)$ o`u $\mathcal{F}$ est un ensemble fini de symboles et $arite$ est une fonction de $\mathcal{F}$ dans $\mathbb{N}$ qui indique l'arit´e des symboles. Pour tout ensemble de variables $\mathcal{X}$, les termes sont d´efinis par : un symbole d'arit´e 0 est un terme, une variable de $\mathcal{X}$ est un terme et si $\mathtt{f}$ est un symbole d'arit´e $n$ et $\mathtt{t}, \ldots, \mathtt{t}_n$ sont des termes, alors $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ est un terme. L'ensemble des termes est not´e $\mathcal{T}(\mathcal{F},\mathcal{X})$ et l'ensemble $\mathcal{T}(\mathcal{F}) = \mathcal{T}(\mathcal{F},\emptyset)$ est l'ensembles des termes clos. Ainsi, les arbres ordonn´es et ´etiquet´es se repr´esentent par des termes. Pour tout terme $\mathtt{t}$, l'ensemble des sous-termes de $\mathtt{t}$ est l'ensemble $\mathrm{sub}(\mathtt{t})$ d´efini par : $\mathtt{t} \in \mathrm{sub}(\mathtt{t})$ et si $\mathtt{f}(\mathtt{t}_1, \ldots, \mathtt{t}_n) \in \mathrm{sub}(\mathtt{t})$ alors $\mathtt{t}_1, \ldots, \mathtt{t}_n$ sont dans $\mathrm{sub}(\mathtt{t})$. Pour tout ensemble de termes $\mathcal{E}$, $\mathrm{sub}(\mathcal{E})$ est la r´eunion des ensembles des sous-termes des ´el´ements de $\mathcal{E}$. Un *contexte* est un terme $\mathtt{c}[\diamond]$ qui contient une variable particuli`ere $\diamond$ `a une seule occurence. La variable $\diamond$ marque un emplacement vide dans

le terme. Ainsi, la substitution de $\diamond$ par un terme $u$ se note $c[u]$. On pourra utiliser cette notation pour indiquer que le terme $u$ est un sous-terme de $c[u]$.

## 2.2 Les automates d'arbres

Un *automate d'arbres* d'´etats fini (AEFN) est un quadruplet $\mathcal{A} = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_F, \underset{\mathcal{A}}{\longrightarrow} \rangle$ où $\mathcal{Q}$ est un ensemble fini d'´etats, $\mathcal{Q}_F \subseteq \mathcal{Q}$ est l'ensemble des ´etats finaux et $\underset{\mathcal{A}}{\longrightarrow}$ est l'ensemble des transitions. Une transition est une règle la forme $f(q_1, \ldots, q_n) \underset{\mathcal{A}}{\longrightarrow} q$ où $q$ et $q_1, \ldots, q_n$ sont des ´etats de $\mathcal{Q}$ et $f$ est un symbole d'arit´e $n$. En particulier, une transition peut être de la forme $a \underset{\mathcal{A}}{\longrightarrow} q$ où $a$ est un symbole d'arit´e nulle.

Un automate d'arbres d'´etats fini est *déterministe* (AEFD) si il n'existe pas deux transition distinctes possédant des parties gauches identiques. Pour ces automates, $\underset{\mathcal{A}}{\longrightarrow}$ est une fonction.

La dérivation simple $\underset{\mathcal{A}}{\longrightarrow}$ est définie par $t \underset{\mathcal{A}}{\longrightarrow} u$ si et seulement si il existe une transition $f(q_1, \ldots, q_n) \underset{\mathcal{A}}{\longrightarrow} q$ telle que $t = v[f(q_1, \ldots, q_n)]$ et $u = v[q]$. On notera que $\underset{\mathcal{A}}{\longrightarrow} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{Q}) \times \mathcal{T}(\mathcal{F}, \mathcal{Q})$, et que les ´etats de $\mathcal{Q}$ sont des symboles d'arit´e nulle.

La relation de dérivation $\overset{*}{\underset{\mathcal{A}}{\longrightarrow}}$ est la clôture réflexive et transitive de $\underset{\mathcal{A}}{\longrightarrow}$. Le langage d'arbres *reconnu* par $\mathcal{A}$ est l'ensemble $\mathcal{L}_{\mathcal{A}} = \{t \in \mathcal{T}(\mathcal{F}) \ : \ t \overset{*}{\underset{\mathcal{A}}{\longrightarrow}} q_f \text{ et } q_f \in \mathcal{Q}_F\}$.

## 2.3 Langages réguliers d'arbres réversibles

**Définition 1**
*Un AEFD $\mathcal{A}$ est* réversible *si et seulement si*

1. *il n'existe pas deux règles distinctes possédant des parties gauches qui ne diffèrent que d'un unique symbole. C'est à dire qu'il n'existe pas deux transitions $f(p_1, \ldots, p_n, q, p_{n+1}, \ldots, p_m) \underset{\mathcal{A}}{\longrightarrow} p$ et $f(p_1, \ldots, p_n, q', p_{n+1}, \ldots, p_m) \underset{\mathcal{A}}{\longrightarrow} p$ avec $q \neq q'$,*
2. *$\mathcal{A}$ possède un unique ´etat final.*

*Un langage d'arbres est* réversible *si et seulement si il est reconnu par un AEFD réversible.*

On remarquera que dans ce cas, un symbole d'arité $n$ et $n-1$ ´etats définissent au plus une transition possible.

**Exemple 2**
*L'AEFD $\mathcal{A}_1 = \langle \{Xor, Not, \mathtt{true}, \mathtt{false}\}, \{q_0, q_1\}, q_1, \underset{\mathcal{A}_1}{\longrightarrow} \rangle$ suivant est réversible.*
$\underset{\mathcal{A}_1}{\longrightarrow} = \{\mathtt{true} \underset{\mathcal{A}_1}{\longrightarrow} q_1,\ \mathtt{false} \underset{\mathcal{A}_1}{\longrightarrow} q_0,\ Not(q_0) \underset{\mathcal{A}_1}{\longrightarrow} q_1,\ Not(q_1) \underset{\mathcal{A}_1}{\longrightarrow} q_0,$

$Xor(q_0,q_0) \underset{\mathcal{A}_1}{\longrightarrow} q_0$, $Xor(q_0,q_1) \underset{\mathcal{A}_1}{\longrightarrow} q_1$, $Xor(q_1,q_0) \underset{\mathcal{A}_1}{\longrightarrow} q_1$, $Xor(q_1,q_1) \underset{\mathcal{A}_1}{\longrightarrow} q_0\}$.
*Le langage régulier d'arbres réversible* $\mathcal{L}_{\mathcal{A}_1}$ *est l'ensemble des expressions booléennes vraies.*

## 2.4 Les grammaires régulières d'arbres réversibles

Une *grammaire régulière d'arbres*, notée GRA, est un quadruplet $\Gamma = \langle \mathcal{F}, \mathcal{X}, \underset{\Gamma}{\rightarrow}, S\rangle$ où $S \in \mathcal{X}$ est la variable de départ. Toutes les productions sont de la forme $X \underset{\Gamma}{\rightarrow} \mathsf{t}$ où $X$ est une variable de $\mathcal{X}$ appelée tête et $\mathsf{t}$ est un terme de $\mathcal{T}(\mathcal{F},\mathcal{X})$. Ainsi, les productions de la forme $X \underset{\Gamma}{\rightarrow} Y$, où $Y$ est une variable de $\mathcal{X}$ ne sont pas autorisées.

Définissons $\mathsf{t} \underset{\Gamma}{\longrightarrow} \mathsf{u}$ par : $\mathsf{t} \underset{\Gamma}{\rightarrow} \mathsf{u}$ si et seulement si il existe une production $X \underset{\Gamma}{\rightarrow} \mathsf{v}$ telle que $\mathsf{u} = \mathsf{t}[X \leftarrow \mathsf{v}]$, où $\mathsf{t}[X \leftarrow \mathsf{v}]$ est le terme obtenu par substitution de $X$ dans $\mathsf{t}$ par $\mathsf{v}$.

La relation de dérivation $\underset{\Gamma}{\overset{*}{\rightarrow}}$ est la clôture réflexive et transitive de $\underset{\Gamma}{\longrightarrow}$ Le langage *produit* par $\Gamma$ est l'ensemble $\mathcal{L}_\Gamma = \{\mathsf{t} \in \mathcal{T}(\mathcal{F}) \ : \ S \underset{\Gamma}{\overset{*}{\rightarrow}} \mathsf{t}\}$.

**Définition 3**
*Une grammaire* $\Gamma$ *est* réversible *si et seulement si :*

1. *il n'existe pas deux productions* $X \underset{\Gamma}{\rightarrow} \mathsf{c}[Y]$ *et* $X \underset{\Gamma}{\rightarrow} \mathsf{c}[Z]$ *avec la même tête* $X$ *et telles que* $Y \neq Z$,
2. *il n'existe pas deux productions* $X \underset{\Gamma}{\rightarrow} \mathsf{t}$ *et* $Y \underset{\Gamma}{\rightarrow} \mathsf{t}$ *avec la même partie droite et telles que* $X \neq Y$.

**Définition 4**
*Une GRA* $\Gamma$ *est* normale *si chaque production est de la forme* $X \underset{\Gamma}{\rightarrow} \mathtt{a}$, *ou de la forme* $X \underset{\Gamma}{\rightarrow} \mathtt{f}(X_1,\ldots,X_n)$.

**Théorème 5**
*Un langage* $\mathcal{L}$ *est réversible si et seulement si il est produit par une GRA réversible et normale.*

**Exemple 6**
*Soit la GRA* $G_1 = \langle\{Xor, Not, \mathtt{true}, \mathtt{false}\}, \{q_0, q_1\}, q_1, \underset{\mathcal{A}_1}{\longrightarrow}\rangle$, $\underset{\mathcal{A}_1}{\longrightarrow} = \{q_1 \underset{G_1}{\longrightarrow} \mathtt{true}$, $q_0 \underset{G_1}{\longrightarrow} \mathtt{false}$, $q_1 \underset{G_1}{\longrightarrow} Not(q_0)$, $q_1 \underset{G_1}{\longrightarrow} Xor(q_0,q_1)$, $q_1 \underset{G_1}{\longrightarrow} Xor(q_1,q_0)$, $q_0 \underset{G_1}{\longrightarrow} Not(q_1)$, $q_0 \underset{G_1}{\longrightarrow} Xor(q_0,q_0)$, $q_0 \underset{G_1}{\longrightarrow} Xor(q_1,q_1)\}$. *Le langage* $\mathcal{L}_{\mathcal{A}_1}$ *est produit par* $G_1$ *qui est réversible et normale.*

## 2.5 Apprentissage des langages réguliers d'arbres réversibles

Une *présentation positive* d'un langage $\mathcal{L}$ est une suite $t_1, \ldots, t_n, \ldots$ qui énumère les éléments de $\mathcal{L}$. Soit $\Omega$ une classe d'automates (ou de grammaires) donnée. Un algorithme d'apprentissage $A$ prend pour entrée, une partie finie $t_1, \ldots, t_n$ d'une présentation positive d'un langage $\mathcal{L}$ et propose un automate $A(t_1, \ldots, t_n)$ en sortie. Un algorithme d'inférence $A$ converge vers $\mathcal{L}$ si il existe une borne $N$ à partir de laquelle le langage produit par $A(t_1, \ldots, t_n)$, $n > N$ est toujours égal à $\mathcal{L}$. Une classe de langages $\mathbb{L}$ est *identifiable* si et seulement si il existe un algorithme d'apprentissage $A$ tel que pour tout langage $\mathcal{L}$ de la classe et toute présentation positive de $\mathcal{L}$, $A$ converge vers $\mathcal{L}$.

**Théorème 7**
*La classe des langages réguliers d'arbres réversibles est identifiable.*

Nous allons démontrer ce théorème dans le prochain chapitre.

# 3 Apprentissage des langages réguliers d'arbres réversibles

## 3.1 Une caractérisation algébrique des langages réguliers d'arbres réversibles

Une relation d'équivalence $\equiv$ est *close par contexte* si pour tous termes $t$ et $u$ dans $\mathcal{T}(\mathcal{F})$, $t \equiv u$ si pour tout contexte $c[\diamond]$, $c[t] \equiv c[u]$. Une congruence $\equiv$ sur $\mathcal{T}(\mathcal{F})$ est une relation d'équivalence close par contexte.

Pour un AEFD donné $\mathcal{A}$, la relation d'équivalence $\equiv_{\mathcal{A}}$ est définie par : $t \equiv_{\mathcal{A}} u$ si $t \xrightarrow[\mathcal{A}]{*} q$ et $u \xrightarrow[\mathcal{A}]{*} q$. On voit facilement que $\equiv_{\mathcal{A}}$ est close par contexte et donc est une congruence.

**Lemme 8**
*Soit $\mathcal{A}$ un AEFD réversible. Pour tout contexte $c[\diamond]$ et pour tous termes $t$ et $u$ tels que $c[t]$ et $c[u]$ soient dans $\mathcal{L}_{\mathcal{A}}$, on a $t \equiv_{\mathcal{A}} u$.*

*Démonstration.* Par induction sur la taille du contexte.
*Cas de base :* Supposons que $c[\diamond] = \diamond$. Par hypothèse, $t$ et $u$ sont dans $\mathcal{L}_{\mathcal{A}}$. Puisque $\mathcal{A}$ est réversible, il existe un unique état final $q_f$ et donc, on a $t \xrightarrow[\mathcal{A}]{*} q_f$ et $u \xrightarrow[\mathcal{A}]{*} q_f$. D'où $t \equiv_{\mathcal{A}} u$.
*Étape d'induction :* Supposons que $c[\diamond] = c'[f(t_1, \ldots, t_n, \diamond, t_{n+1}, \ldots, t_m)]$. Par hypothèse, il existe un état $q_i$ avec $t_i \xrightarrow[\mathcal{A}]{*} q_i$. Puisque $\mathcal{A}$ est réversible, les états $q_1, \ldots, q_m$ déterminent une unique transition dont la partie gauche est $c'[f(q_1, \ldots, q_n, q, q_{n+1}, \ldots, q_m)]$. Ceci implique $t \xrightarrow[\mathcal{A}]{*} q$ et $u \xrightarrow[\mathcal{A}]{*} q$, donc $t \equiv_{\mathcal{A}} u$.

□

Pour tout langage d'arbres r´egulier $\mathcal{L}$, la congruence $\equiv_{\mathcal{L}}$ est d´efi nie par : $t \equiv_{\mathcal{L}} u$ si pour tout contexte $c[\diamond]$, $c[t] \in \mathcal{L}$ si et seulement si $c[u] \in \mathcal{L}$. D'apr`es le th´eor`eme de Myhill-Nerode, pour tout automate $\mathcal{A}$ qui reconnaît $\mathcal{L}$, le nombre de classes de $\equiv_{\mathcal{L}}$ est inf´erieur ou ´egal au nombre de classes de $\equiv_{\mathcal{A}}$.

En cons´equence, l'AEFD minimal (relativement au nombre d'´etats) $\mathcal{A}_{\mathcal{L}} = \langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_F, \xrightarrow[\mathcal{L}]{} \rangle$ qui reconnaît $\mathcal{L}$ est d´efi ni comme suit. Soit $[t]$ la classe d'´equivalence de $t$ pour $\equiv_{\mathcal{L}}$. Posons alors $\mathcal{Q} = \{[t] : t \in \text{sub}(\mathcal{L})\}$ et $\mathcal{Q}_F = \{[t] : t \in \mathcal{L}\}$. Pour chaque ´etat $[t_1], \ldots, [t_n]$ et pour chaque symbole n-aire $f \in \mathcal{F}$, il existe une transition $f([t_1], \ldots, [t_n]) \xrightarrow[\mathcal{L}]{} [f(t_1, \ldots, t_n)]$.

**Théorème 9**
*Un langage r´egulier d'arbres $\mathcal{L}$ est r´eversible si et seulement si pour tout contexte $c[\diamond]$ et pour tous termes $t$ et $u$, si $c[t]$ et $c[u]$ sont dans $\mathcal{L}$ alors on a $t \equiv_{\mathcal{L}} u$.*

*Démonstration.* Il existe un AEFD r´eversible $\mathcal{A}$ qui reconnaît $\mathcal{L}$. D'apr`es le th´eor`eme de Myhill-Nerode, si $t \equiv_{\mathcal{A}} u$ alors $t \equiv_{\mathcal{L}} u$. Le lemme 8 permet de conclure.

Inversement, consid´erons le AEFD minimal $\mathcal{A}_{\mathcal{L}}$ qui reconnaît $\mathcal{L}$. $\mathcal{A}_{\mathcal{L}}$ poss`ede un unique ´etat fi nal car par hypoth`ese, tous les arbres de $\mathcal{L}$ appartiennent `a la même classe d'´equivalence.

Supposons maintenant l'existence de deux transitions de la forme $f(p_1, \ldots, p_n, q, p_{n+1}, \ldots, p_m) \xrightarrow[\mathcal{L}]{} q''$ et $f(p_1, \ldots, p_n, q', p_{n+1}, \ldots, p_m) \xrightarrow[\mathcal{L}]{} q''$. Puisque $\mathcal{A}_{\mathcal{L}}$ est minimal, il existe des termes $t_1, \ldots, t_m$ tels que $t_i \xrightarrow[\mathcal{L}]{*} p_i$. Ceci permet de consid´erer le contexte $c[\diamond] = C[f(t_1, \ldots, t_n, \diamond, t_{n+1}, \ldots, t_m)]$ tel que $c[q] \xrightarrow[\mathcal{L}]{*} q_f$ et $c[q'] \xrightarrow[\mathcal{L}]{*} q_f$, o`u $q_f$ est l'´etat fi nal de $\mathcal{A}$. Alors, par hypoth`ese, on a $q = q'$ et donc $\mathcal{A}_{\mathcal{L}}$ est r´eversible. On conclut que $\mathcal{L}$ est r´eversible.

□

La d´emonstration ci-dessus induit les cons´equences suivantes.

**Corollaire 10**
*Un langage r´egulier d'arbres est r´eversible si et seulement si l'AEFD minimal $\mathcal{A}_{\mathcal{L}}$ est r´eversible.*

**Exemple 11**
*Soit $\mathcal{L} = \{f(g^n(a)) : n \geq 0\} \cup \{g(f^n(a)) : n \geq 0\}$. Le langage r´egulier d'arbres $\mathcal{L}$ est reconnu par l'AEFD $\mathcal{A}$ dont les transitions sont :*

*$a \xrightarrow[\mathcal{A}]{} A$, $g(A) \xrightarrow[\mathcal{A}]{} B$, $f(A) \xrightarrow[\mathcal{A}]{} C$, $g(B) \xrightarrow[\mathcal{A}]{} G$, $f(C) \xrightarrow[\mathcal{A}]{} S$, $g(G) \xrightarrow[\mathcal{A}]{} G$, $f(F) \xrightarrow[\mathcal{A}]{} F$, $g(F) \xrightarrow[\mathcal{A}]{} S$, $f(G) \xrightarrow[\mathcal{A}]{} S$. Les ´etats fi naux sont $S$,$B$ et $C$.*

*L'AEFD $\mathcal{A}$ est minimal et donc isomorphe `a $\mathcal{A}_{\mathcal{L}}$. En cons´equence du corollaire 10, $\mathcal{L}$ n'est pas un langage r´egulier d'arbres r´eversible (bien qu'union de deux langages r´eguliers d'arbres r´eversibles).*

## 3.2 Ensembles caractéristiques

Soit $\mathcal{L}$ un langage r´egulier d'arbres r´eversible reconnu par l'automate minimal $\mathcal{A}_{\mathcal{L}}$. On associe chaque ´etat $q$ `a un terme $\mathsf{rt}(q)$ de $\mathcal{T}(\mathcal{F})$ tel que $\mathsf{rt}(q) \xrightarrow[\mathcal{L}]{*} q$ et `a un contexte minimale $\mathsf{C}_q[\diamond]$ (relativement `a la taille), tel que $\mathsf{C}_q[q] \xrightarrow[\mathcal{L}]{*} q_f$.

L'ensemble d'*exemples caractéristiques* $CS(\mathcal{L})$ est le plus petit ensemble qui contient

1. le terme $\mathsf{C}_q[\mathsf{rt}(q)]$ pout tout ´etat $q$,
2. le terme $\mathsf{C}_q[\mathsf{f}(\mathsf{rt}(q_1),\ldots,\mathsf{rt}(q_n))]$ pour toute transition $\mathsf{f}(q_1,\ldots,q_n) \xrightarrow[\mathcal{L}]{} q$.

On voit facilement que $CS(\mathcal{L}) \subseteq \mathcal{L}$. On remarquera ´egalement que $\mathsf{C}_{q_f}[\diamond] = \diamond$, si $q_f$ est un ´etat final.

**Théorème 12**
*Soient $\mathcal{L}_1$ et $\mathcal{L}_2$ deux langages r´eversibles sur $\mathcal{T}(\mathcal{F})$ et supposons que $CS(\mathcal{L}_1) \subseteq \mathcal{L}_2$. Alors $\mathcal{L}_1 \subseteq \mathcal{L}_2$.*

*Démonstration.* Supposons que $\mathcal{A}' = \langle \mathcal{F}, \mathcal{Q}_1, \{q_{f_1}\}, \xrightarrow[1]{} \rangle$ et $\mathcal{A}_2 = \langle \mathcal{F}, \mathcal{Q}_2, \{q_{f_2}\}, \xrightarrow[2]{} \rangle$ soient les AEFD minimaux qui reconnaissent respectivement $\mathcal{L}_1$ et $\mathcal{L}_2$.

Pour tout ´etat $q \in \mathcal{Q}$, $\mathsf{C}_q[\mathsf{rt}(q)] \in CS(\mathcal{L}_1) \subseteq \mathcal{L}_2$. Alors, $\mathsf{rt}(q) \in \mathrm{sub}(\mathcal{L}_2)$ et il existe une unique fonction $\theta : \mathcal{Q}_1 \to \mathcal{Q}_2$ telle que $\mathsf{rt}(q) \xrightarrow[2]{*} \theta(q)$, car $\mathcal{A}_2$ est d´eterministe.

Le th´eor`eme 9, nous permet de faire une observation importante. Pour tout ´etat $\mathsf{t} \in \mathcal{T}(\mathcal{F})$, si $\mathsf{C}_q[\mathsf{t}]$ est dans $\mathcal{L}_2$, alors $\mathsf{t} \xrightarrow[2]{*} \theta(q)$.

Nous montrons par induction sur la taille du terme $\mathsf{t} \in \mathcal{T}(\mathcal{F})$, que pour chaque ´etat $q \in \mathcal{Q}$, si $\mathsf{C}_q[\mathsf{t}] \in \mathcal{L}_1$ alors $\mathsf{C}_q[\mathsf{t}] \in \mathcal{L}_2$. En cons´equence, puisque $\mathsf{C}_{q_{f_1}}[\mathsf{t}] = \mathsf{t}$, si $\mathsf{t} \in \mathcal{L}_1$ alors $\mathsf{t} \in \mathcal{L}_2$.

*Cas de base :* Supposons que $\mathsf{t}$ est un symbole d'arit´e nulle. Alors il existe une transition $\mathsf{t} \xrightarrow[1]{} q$ et $\mathsf{C}_q[\mathsf{t}] \in CS(\mathcal{L}_1) \subseteq \mathcal{L}_2$.

*Étape d'induction :* Supposons que $\mathsf{t} = \mathsf{f}(\mathsf{t}_1,\ldots,\mathsf{t}_n)$. Par hypoth`ese, il existe un ´etat $q_i$ dans $\mathcal{Q}_1$, tel que $\mathsf{t}_i \xrightarrow[1]{*} q_i$. Puisque $\mathsf{C}_{q_i}[\mathsf{t}_i] \in \mathcal{L}_1$, nous pouvons appliquer l'hypoth`ese d'induction pour obtenir $\mathsf{C}_{q_i}[\mathsf{t}_i] \in \mathcal{L}_2$. De plus, $\mathsf{t}_i \xrightarrow[2]{*} \theta(q_i)$. Nous avons,

$$\mathsf{C}_q[\mathsf{f}(\mathsf{t}_1,\ldots,\mathsf{t}_n)] \xrightarrow[2]{*} \mathsf{C}_q[\mathsf{f}(\theta(q_1),\ldots,\theta(q_n))] \tag{1}$$

et d'autre part, puisque $\mathsf{C}_q[\mathsf{f}(\mathsf{rt}(q_1),\ldots,\mathsf{rt}(q_n))] \in \mathcal{L}_2$, l'on a

$$\mathsf{C}_q[\mathsf{f}(\mathsf{rt}(q_1),\ldots,\mathsf{rt}(q_n))] \xrightarrow[2]{*} \mathsf{C}_q[\mathsf{f}(\theta(q_1),\ldots,\theta(q_n))] \tag{2}$$

$$\xrightarrow[2]{*} q_{f_2} \tag{3}$$

**R1** SI $\mathtt{f}(p_1,\ldots,p_n,q,p_{n+1},\ldots,p_m) \to p$ et $\mathtt{f}(p_1,\ldots,p_n,q',p_{n+1},\ldots,p_m) \to p$ alors $q = q'$.

**R2** SI $\mathtt{f}(q_1,\ldots,q_n) \to q$ et $\mathtt{f}(q_1,\ldots,q_n) \to q'$ alors $q = q'$.

**R3** si $q_{f_1}$ et $q_{f_2}$ sont deux ´etats fi naux alors $q_{f_1} = q_{f_2}$.

FIG. 1 – *Les règles de réduction*

En combinant (1) et (3), on obtient

$$\mathtt{C}_q[\mathtt{f}(\mathtt{t}_1,\ldots,\mathtt{t}_n)] \xrightarrow[2]{*} q_{f_2} \tag{4}$$

et donc $\mathtt{C}_q[\mathtt{t}] \in \mathcal{L}_2$.

□

D'apr`es Angluin (Angluin, 1982), les exemples caract´eristiques sont des ensembles *telltale* ; cette observation nous permet de conclure d`es `a pr´esent, que la classe des langages r´eguliers d'arbres r´eversibles est identifi able.

## 3.3 Un algorithme d'apprentissage effi cace

L'algorithme d'apprentissage fonctionne comme suit. L'entr´ee est un ensemble fi ni $S$, d'exemples positifs, c'est `a dire d'´el´ements d'un langage inconnu.

Commençons par d´efi nir l'automate d'arbres pr´efi xe $\mathrm{PTA}(S) = \langle \mathcal{F}_0, \mathcal{Q}, \mathcal{Q}_{F_0}, \xrightarrow[0]{} \rangle$. Pour chaque sous-terme $\mathtt{t}$ de $\mathrm{sub}(S)$, il existe un ´etat not´e $[\mathtt{t}]$ dans $\mathcal{Q}$. L'ensemble des ´etats fi naux $\mathcal{Q}_{F_0}$ contient chaque ´etat $[\mathtt{t}]$, o`u $\mathtt{t} \in S$. Pour chaque sous-terme $\mathtt{f}(\mathtt{t}_1,\ldots,\mathtt{t}_n)$ de $\mathrm{sub}(S)$, il existe une transition $\mathtt{f}([\mathtt{t}_1],\ldots,[\mathtt{t}_n]) \xrightarrow[0]{} [\mathtt{f}(\mathtt{t}_1,\ldots,\mathtt{t}_n)]$. On voit facilement que $\mathcal{L}_{\mathrm{PTA}(S)} = S$.

Puis, une suite $\mathcal{A}_0 = \mathrm{PTA}(S), \mathcal{A}_1, \ldots, \mathcal{A}_n$ d'AEFN est calcul´ee par applications successives d'une des r`egles de la fi gure 1, jusqu'`a ce qu'aucune r`egle ne puisse plus s'appliquer. On remarque qu'un automate $\mathcal{A}_{i+1}$ est obtenu `a partir d'un automate $\mathcal{A}_i$ par assimilation de deux ´etats distincts de $\mathcal{A}_i$. Le processus termine apr`es $n$ ´etapes, puisque `a chaque ´etape, le nombre d'´etats diminue strictement. On note $\mathbf{nf}(S) = \mathcal{A}_n$.

**Lemme 13**
*L'AEFD* $\mathbf{nf}(S)$ *est r´eversible.*

*Démonstration.* Le processus s'arrête si aucune r`egle n'est applicable, ce qui implique que $\mathbf{nf}(S)$ est r´eversible.

□

Un *homomorphisme d'automates* entre les AEFN $\mathcal{A}' = \langle \mathcal{F}, \mathcal{Q}', \mathcal{Q}_{F'}, \xrightarrow[']{} \rangle$ et $\mathcal{A}_2 = \langle \mathcal{F}, \mathcal{Q}_2, \mathcal{Q}_{F_2}, \xrightarrow[2]{} \rangle$ est un fonction $\theta$ qui applique $\mathcal{Q}_1$ `a $\mathcal{Q}_2$ et qui a la propri´et´e suivante : si

$\mathtt{f}(q_1,\ldots,q_n) \underset{1}{\rightarrow} q$ alors $\mathtt{f}(\theta(q_1),\ldots,\theta(q_n)) \underset{2}{\rightarrow} \theta(q)$, pour chaque transition de $\mathcal{A}'$ et $\theta(q_f) \in \mathcal{Q}_{F_2}$ si $q_f \in \mathcal{Q}_{F_1}$.

On dit que $\mathcal{A}' \sqsubseteq \mathcal{A}_2$ si il existe un homomorphisme d'automates de $\mathcal{A}'$ vers $\mathcal{A}_2$ On remarque que si $\mathcal{A}' \sqsubseteq \mathcal{A}_2$ alors $\mathcal{L}_{\mathcal{A}'} \subseteq \mathcal{L}_{\mathcal{A}_2}$.

**Lemme 14**
*Si $S \subseteq \mathcal{L}$ alors PTA$(S) = \mathcal{A}_0 \sqsubseteq \mathcal{A}_{\mathcal{L}}$.*

*Démonstration.* D´efi nissons $\theta$ comme suit. $\theta([\mathtt{t}])$ est la classe d'´equivalence de $\mathtt{t}$.

□

**Lemme 15**
*Soit $\mathcal{A}$ un AEFD r´eversible. Supposons que $\mathcal{A}_{i+1}$ par application d'une des r`egles* $\mathbf{R1}$, $\mathbf{R2}$ *o`u* $\mathbf{R3}$ *`a $\mathcal{A}$ Si $\mathcal{A}_i \sqsubseteq \mathcal{A}$ alors $\mathcal{A}_{i+1} \sqsubseteq \mathcal{A}$.*

*Démonstration.* Soit $\theta$ l'homomorphisme d'automates de $\mathcal{A}_i$ vers $\mathcal{A}$. Il y a trois cas possibles.
La r`egle $\mathbf{R1}$ a produit $\mathcal{A}_{i+1}$. Puisque $\mathcal{A}$ est r´eversible, on a $\theta(q) = \theta(q')$.
La r`egle $\mathbf{R2}$ a produit $\mathcal{A}_{i+1}$. Puisque $\mathcal{A}$ est d´eterministe, on a $\theta(q) = \theta(q')$.
La r`egle $\mathbf{R3}$ a produit $\mathcal{A}_{i+1}$. Puisque $\mathcal{A}$ poss`ede un unique ´etat fi nal, on a $\theta(q_{f_1}) = \theta(q_{f_2})$.

Nous concluons que $\theta$ est un homomorphisme d'automates de $\mathcal{A}_{i+1}$ vers $\mathcal{A}$.

□

**Lemme 16**
*Pour tout langage r´eversible $\mathcal{L}$, si $S \subseteq \mathcal{L}$ alors $\mathcal{L}_{\mathbf{nf}(S)} \subseteq \mathcal{L}$.*

*Démonstration.* Cons´equence des lemmes 15 et 14.

□

**Exemple 17**
*Consid´erons les donn´ees $\{f(a,b), f(g(a),b), f(g(g(a)),b)\}$. L'algorithme commence par calculer l'automate PTA$(S)$ comme :*

*PTA$(S) = \langle \mathcal{F}, \mathcal{Q}_0, \mathcal{Q}_{F_0}, \underset{0}{\rightarrow} \rangle$, o`u :*

- *$\mathcal{F}$ est l'ensemble $\{f, g, a, b\}$,*
- *$\mathcal{Q}_0$ est l'ensemble $\{[a], [b], [f(a,b)], [g(a)], [f(g(a),b)], [g(g(a))], [f(g(g(a)),b)]\}$,*
- *$\mathcal{Q}_{F_0}$ est l'ensemble $\{[f(a,b)], [f(g(a),b)], [f(g(g(a)),b)]\}$,*
- *$\underset{0}{\rightarrow}$ est l'ensemble $\{a \underset{0}{\rightarrow} [a],\ b \underset{0}{\rightarrow} [b],\ f([a],[b]) \underset{0}{\rightarrow} [f(a,b)],\ g([a]) \underset{0}{\rightarrow} [g(a)],\ f([g(a)],[b]) \underset{0}{\rightarrow} [f(g(a),b)],\ g([g(a)]) \underset{0}{\rightarrow} [g(g(a))],\ f([g(g(a))],[b]) \underset{0}{\rightarrow} [f(g(g(a)),b)]\}$.*

*En appliquant trois fois la r`egle* $\mathbf{R3}$*, on obtient :* $[f(a,b)] = [f(g(a),b)] = [f(g(g(a)),b)]$ *et le nouvel ensemble* $\{a \rightarrow [a],\ b \rightarrow [b],\ f([a],[b]) \rightarrow [f(a,b)],\ g([a]) \rightarrow [g(a)],\ f([g(a)],[b]) \rightarrow [f(a,b)],\ g([g(a)]) \rightarrow [g(g(a))],\ f([g(g(a))],[b])$

$\rightarrow [f(a,b)]\}$. *Nous appliquons maintenant deux fois la r`egle* $\mathbf{R1}$ *pour conclure* $[a] = [g(a)] = [g(g(a))]$ *et produire l'automate final :* $\langle \mathcal{F}, \mathcal{Q}, \mathcal{Q}_F, \rightarrow \rangle$, *o`u :*

- $\mathcal{F}$ *est l'ensemble* $\{f,g,a,b\}$,
- $\mathcal{Q}$ *est l'ensemble* $\{[a], [b], [f(a,b)]\}$,
- $\mathcal{Q}_F$ *est l'ensemble* $\{[(f(a,b)]\}$,
- $\rightarrow$ *est l'ensemble* $\{a \rightarrow [a], b \rightarrow [b], f([a],[b]) \rightarrow [f(a,b)], g([a]) \rightarrow [a]\}$.

**Théorème 18 (Démonstration du théorème 7)**
*Supposons que* $\mathcal{L}$ *est un langage r´egulier d'arbres r´eversible. Pour chaque pr´esentation d'exemples positifs* $\mathtt{t}_1, \ldots, \mathtt{t}_n, \ldots$ *de* $\mathcal{L}$, *il existe une ´etape* $N$ *telle que pour tout* $n > N$, $\mathbf{nf}(\mathtt{t}_1, \ldots, \mathtt{t}_n)$ *est l'AEFD r´eversible minimal qui reconnaît* $\mathcal{L}$. *En d'autres termes, la classe des langages r´eguliers d'arbres r´eversible est identifiable.*

*Démonstration.* Il existe une ´etape $N$ telle que pour tout $n > N$, $\mathtt{t}_1, \ldots, \mathtt{t}_n$ contient $CS(\mathcal{L})$. D'apr`es le lemme 16, $\mathcal{L}_{\mathbf{nf}(\mathtt{t}_1,\ldots,\mathtt{t}_n)} \subseteq \mathcal{L}$. Or $\mathcal{L}_{\mathbf{nf}(\mathtt{t}_1,\ldots,\mathtt{t}_n)}$ est un langage r´eversible. En appliquant le th´eor`eme 12, on conclut que $\mathcal{L} \subseteq_{\mathbf{nf}(\mathtt{t}_1,\ldots,\mathtt{t}_n)} \mathcal{L}_{\mathbf{nf}(\mathtt{t}_1,\ldots,\mathtt{t}_n)}$. Nous avons donc $\mathcal{L} = \mathcal{L}_{\mathbf{nf}(\mathtt{t}_1,\ldots,\mathtt{t}_n)}$.

□

L'algorithme d'apprentissage est incr´ementale et s'ex´ecute en temps quadratique en fonction de la taille des donn´ees d'entr´ee.

# 4 APPRENDRE À PARTIR D'EXEMPLES STRUCTURÉS

Nous donnons dans ce chapitre une illustration de notre m´ethode d'apprentissage en consid´erant l'apprentissage `a la limite des langages de mots.

## 4.1 Langages algébriques

Rappelons bri`evement la d´efinition des grammaires alg´ebriques (CFG). Une grammaire alg´ebrique $G$ est un quadruplet $\langle \Sigma, \mathcal{N}, \underset{G}{\rightarrow}, S \rangle$ o`u $\Sigma$ est l'alphabet, $\mathcal{N}$ est l'ensemble des symboles non-terminaux, et $S \in \mathcal{N}$ est le symbole de d´epart. $\Sigma^*$ est l'ensemble des mots g´en´er´es sur l'alphabet $\Sigma$. ~~est~~ l'ensemble des r`egles de production de la forme : $X \underset{G}{\rightarrow} w$ o`u $X \in \mathcal{N}$ et $w \in (\Sigma \cup \mathcal{N})^*$. Le langage $\mathcal{L}_G$ d´efini par $G$ est l'ensemble $\mathcal{L}_G = \{w \in \Sigma^* : S \overset{*}{\underset{G}{\rightarrow}} w\}$. $\mathcal{D}(G)$ est l'ensemble des arbres de d´erivation de $G$.

Il existe une ´etroite relation entre les langages alg´ebriques et les grammaires r´eguli`eres d'arbres. Pour commencer, le langage de mots obtenu `a partir des arbres d'un langage r´egulier d'arbres, en lisant pour chaque arbre, le mot form´e par les ´etiquettes des feuilles, est un langage alg´ebrique. En effet, soit $\Sigma$ l'ensemble des symboles d'arit´e nulle de $\mathcal{F}$. On d´efinit la fonction **yield** de $\mathcal{T}(\mathcal{F})$ dans $^*\Sigma$

comme suit : **yield**(a) = a et **yield**(f(t$_1$,...,t$_n$)) = **yield**(t$_1$)...**yield**(t$_n$). Si $\mathcal{L} \subseteq \mathcal{T}(\mathcal{F})$ est un langage d'arbres, alors **yield**($\mathcal{L}$) = {**yield**(t) : t $\in \mathcal{L}$}.

**Théorème 19**
*Si $\mathcal{L}$ est un langage r´egulier d'arbres alors **yield**($\mathcal{L}$) est un langage alg´ebrique.*

Ensuite, l'ensemble des arbres de d´erivation d'une grammaire alg´ebrique est un langage r´egulier d'arbres.

**Théorème 20**
*Soit $G$ une grammaire alg´ebrique. L'ensemble $\mathcal{D}(G)$ des arbres de d´erivation de $G$ est un langage r´egulier d'arbres.*

Construisons une GRA $\Gamma = \langle \mathcal{F}, \mathcal{X}, \underset{\Gamma}{\rightarrow}, S \rangle$, `a partir d'une grammaire $G = \langle \Sigma, \mathcal{N}, \underset{G}{\rightarrow}, S \rangle$. À chaque lettre de $\Sigma$ correspond un symbole d'arit´e nulle de $\mathcal{F}$. À chaque r`egle $X \underset{G}{\rightarrow} w$, o`u $w$ est un mot de taille $n$, correspond un symbole $X_n$ de $\mathcal{F}$ d'arit´e $n$. Tout non-terminal de $\mathcal{N}$ est une variable de $\mathcal{X}$. `a chaque production $X \underset{G}{\rightarrow} w_1, \ldots, w_n$, correspond une production $X \underset{\Gamma}{\rightarrow} X_n(Y_1, \ldots, Y_n)$, o`u $Y_i = w_i$ si $w_i$ est dans $\mathcal{N}$ et $Y_i = [w_i]$ sinon. Enfi n, pour chaque lettre a $\in \Sigma$, on a [a] $\underset{\Gamma}{\rightarrow}$ a.

## 4.2 Exemples structurés

Dans le cas des langages alg´ebriques, le probl`eme d'inf´erence d'une grammaire compatible avec un langage, `a partir de la simple pr´esentation d'une s´equence d'´el´ements de ce langage est diffi cile. Ainsi, plusieurs travaux ont introduit l'id´ee d'apprentissage, non plus `a partir d'´el´ements d'un langage, mais `a partir de structures int´egrant de l'information sur la mani`ere dont ont ´et´e produits ces ´el´ements. On parle alors d'*exemples structurés* et l'on passe d'un probl`eme d'apprentissage de langages `a un probl`eme d'apprentissage de grammaires. Citons par exemple, les travaux de Sakakibara (Sakakibara, 1992) pour les grammaires alg´ebriques et de Kanazawa (Kanazawa, 1998) pour les grammaires cat´egorielles. Consid´erons deux cas :

1. une pr´esentation compl`ete d'une CFG $G$ est une suite t$_1$, ..., t$_n$, ... d'arbres de d´erivation de $G$,
2. une fonction de d´es´etiquetage $sk$ est une fonction d´efi nie par : $sk$(a) = a et $sk$(f(t$_1$,...,t$_n$)) = $\sigma_n(sk(\mathrm{t}_1), \ldots, sk(\mathrm{t}_n))$ Une pr´esentation squelette d'une grammaire $G$ est une suite $sk(\mathrm{t}_1), \ldots,$ $sk(\mathrm{t}_n), \ldots$ de l'ensemble des arbres de d´erivation d´es´etiquet´es de $G$.

Une donn´ee d'apprentissage peut donc être un langage r´egulier d'arbres, image par homomorphisme de l'ensemble des arbres de d´erivation d'une grammaire de mots. Nous pouvons alors consid´erer une $h$-pr´esentation d'une CFG $G$ d´efi nie comme suit. Soit $h$ un homomorphisme d'arbres, une $h$-pr´esentation d'une CFG

$G$ est une suite $h(\mathtt{t}_1),\ldots,h(\mathtt{t}_n),\ldots$ , o`u $\mathtt{t}_1,\ldots,\mathtt{t}_n,\ldots$ est une ´enum´eration de l'ensemble des arbres de d´erivation de $G$.

## 4.3 Identification de grammaires

Soient $h(\mathcal{D}(G)) = \{h(\mathtt{t}) : \mathtt{t} \in \mathcal{D}(G)\}$, $\Omega$ une classe de grammaires et $h$ un homomorphisme d'arbres. Pour toute $h$-pr´esentation $h(\mathtt{t}_1),\ldots,h(\mathtt{t}_n),\ldots$, l'algorithme d'apprentissage $A$ *converge vers* $G \in \Omega$ si il existe une borne $N$ telle que pour tout $n > N$, le langage produit $A(h(\mathtt{t}_1),\ldots,h(\mathtt{t}_n))$ soit ´egal `a $h(\mathcal{D}(G))$.

Une classe $\Omega$ de grammaires est *identifiable* `a partir de $h$-pr´esentations si et seulement si il existe un algorithme d'apprentissage $A$ tel que pour toute grammaire $G$ et toute $h$-pr´esentation de celle-ci, $A$ converge vers $G$.

## 4.4 Les travaux de Sakakibara

Sakakibara a d´emontr´e que la classe des grammaires alg´ebriques r´eversibles est identifiable `a partir de pr´esentations squelettes.

**Définition 21**
*Une CFG est r´eversible si et seulement si*

**(Reset-free)** *il n'y a pas deux productions* $X \underset{G}{\rightarrow} wYv$ *et* $X \underset{G}{\rightarrow} wZv$ *telles que* $Y$,$Z$ *soient des non-terminaux et* $Y \neq Z$,

**(Déterministe)** *il n'y a pas deux productions distinctes* $X \underset{G}{\rightarrow} \mathtt{t}$ *et* $Y \underset{G}{\rightarrow} \mathtt{t}$ *poss´edant la m^eme partie droite.*

*Un langage alg´ebrique r´eversible est un langage produit par une CFG r´eversible.*

**Théorème 22 (Sakakibara (Sakakibara, 1992))**
*La classe des grammaires alg´ebriques r´eversibles est identifiable `a partir de pr´esentations squelettes.*

*Démonstration.* Les arbres d'une pr´esentation squelette est un langage r´egulier d'arbres r´eversible. Le th´eor`eme 7 implique donc que cette classe est identifiable. □

## 4.5 L'apprentissage des langages algébriques de mots réversibles

**Définition 23**
*La classe $\mathcal{R}$ des langages alg´ebriques de mots est le plus petit ensemble tel que pour tout langage $\mathcal{L}$ de $\mathcal{R}$, il existe un langage r´egulier d'arbres r´eversible $\mathcal{L}$avec $\mathcal{L} = \textbf{\textit{yield}}(\mathcal{L}_1)$. On dit alors que $\mathcal{L}_1$ est une pr´esentation d'arbres r´eversibles de $\mathcal{L}$.*

Le th´eor`eme 7 nous permet d'´etablir le r´esultat suivant.

**Théorème 24**
*La classe $\mathcal{R}$ est identifiable `a partir de pr´esentations d'arbres r´eversibles.*

Afi n d'illustrer ce r´esultat et de le comparer avec les travaux de Sakakibara nous donnons des exemples.

**Exemple 25**
*La grammaire* $\Gamma$ *dont les productions suivent est r´eversible.* $S \underset{\Gamma}{\rightarrow} f(X,Y)$, $X \underset{\Gamma}{\rightarrow} g(A,X,B)$, $X \underset{\Gamma}{\rightarrow} c$, $Y \underset{\Gamma}{\rightarrow} g(A,Y,B)$, $Y \underset{\Gamma}{\rightarrow} d$, $A \underset{\Gamma}{\rightarrow} a$, $B \underset{\Gamma}{\rightarrow} b$. *D'apr`es le th´eor`eme 7,* $\mathcal{L}_\Gamma$ *est identifiable.*

*De plus, le th´eor`eme pr´ec´edent implique que le langage alg´ebrique* $\boldsymbol{yield}(\mathcal{L}_\Gamma)$ *est identifiable `a partir de toute ´enum´eration de* $\mathcal{L}_\Gamma$ *qui constitue un langage r´egulier d'arbres r´eversible. N´eanmoins,* $\mathcal{L}_\Gamma$ *n'est pas l'ensemble des arbres de d´erivation d'un langage alg´ebrique. Nous avons* $g \overset{*}{\rightarrow} acb$ *et* $g \overset{*}{\rightarrow} adb$ *et, d'autre part nous devons avoir la r`egle* $f \rightarrow gg$. *On devrait donc avoir* $acbacb$ *dans* $\boldsymbol{yield}(\mathcal{L}_\Gamma)$, *ce qui n'est pas le cas.*

**Exemple 26**
*Ce deuxi`eme exemple montre que l'apprentissage peut ˆetre facilit´e par la pr´esentation des arbres de d´erivation complets. La grammaire* $G$ *suivante n'est pas r´eversible car* $G$ *n'est pas d´eterministe.* $S \underset{G}{\rightarrow} ab$, $S \underset{G}{\rightarrow} aXb$, $X \underset{G}{\rightarrow} ab$. *L'ensemble* $\mathcal{D}(G)$ *des arbres de d´erivation est un langage r´eversible. En effet, il peut ˆetre g´en´er´e par la grammaire r´eversible minimale suivante.* $S \underset{G}{\rightarrow} S_2([a],[b])$, $S \underset{G}{\rightarrow} S_3([a],X,[b])$, $X \underset{G}{\rightarrow} X_2([a],[b])$, $[a] \underset{G}{\rightarrow} a$, $[b] \underset{G}{\rightarrow} b$.

## 5 LES LANGAGES D'ARBRES DE DÉPENDANCES

Suivant Dikovsky et Modina (Dikovsky & Modina, 2000), nous pr´esentons une classe des langages d'arbres de d´ependances r´eversibles introduite par Hays (Hays, 1961) et Gaifman (Gaifman, 1965). Une *grammaire de dépendances lexicalisée* (LDG) $\Gamma$ est un quadruplet $\langle \Sigma,N,P,S \rangle$, o`u $\Sigma$ est un alphabet, $N$ est l'ensemble des non-terminaux, $S \in N$ est le symbole de d´epart et chaque r`egle de production

de $P$ est de la forme $X \rightarrow U_1 \ldots U_p\ a\ V_1 \ldots V_q$, o`u $X \in N$ et chaque $U_i$ et $V_j$ est dans $\Sigma \cup N$. La partie droite peut être vue `a la fois comme un arbre ´etiquet´e ordonn´e de hauteur 1, dont la racine est $a$, ou comme un mot $U_1 \ldots U_p a V_1 \ldots V_q$. Ainsi, les nœuds des arbres sont totalement ordonn´es.

Les *arbres de dépendances partiels* sont d´efi nis comme suit.

1. $S$ est un arbre de d´ependances partiel g´en´er´e par $\Gamma$.

2. Si $\ldots X \ldots \mathrm{b} \ldots$ est un arbre de d´ependances partiel g´en´er´e par $\Gamma$, et si

$X \to U_1 \ldots U_p\ a\ V_1 \ldots V_q$ est une production de $\Gamma$, alors

$\ldots U_1 \ldots U_p\ a\ V_1 \ldots V_p \ldots \mathrm{b} \ldots$

est un arbre de d´ependances partiel g´en´er´e par $\Gamma$.

Un *arbre de dépendances* g´en´er´e par une LDG $\Gamma$ est un arbre de d´ependances partiel g´en´er´e par $\Gamma$ dont tous les nœuds sont ´etiquet´es par des symboles terminaux Le langage $\mathcal{D}(\Gamma)$ est l'ensemble des arbres de d´ependances g´en´er´es par $\Gamma$.

On note $\boldsymbol{\alpha}\ a\ \boldsymbol{\beta}$ pour $\alpha_1 \ldots \alpha_p\ a\ \beta_1 \ldots \beta_q$.

**Définition 27**
*Une grammaire LDG* $\Gamma$ *est* r´eversible *si et seulement si les conditions suivantes sont v´erifi´ees.*

1. *Si* $X \to \mathbf{U}\ a\ \mathbf{V}$ *et si* $Y \to \mathbf{U}\ a\ \mathbf{V}$, *alors* $X = Y$.
2. *Si* $X \to \boldsymbol{\alpha}\ Y \boldsymbol{\beta}\ a\ \boldsymbol{\gamma}$ *et si* $X \to \boldsymbol{\alpha}\ Z \boldsymbol{\beta}\ a\ \boldsymbol{\gamma}$,
   *alors* $Y = Z$, *o`u* $Y,Z \in N$.
3. *Si* $X \to \boldsymbol{\alpha}\ a\ \boldsymbol{\beta}\ Y\ \boldsymbol{\gamma}$ *et si* $X \to \boldsymbol{\alpha}\ a\ \boldsymbol{\beta}\ Z\ \boldsymbol{\gamma}$,
   *alors* $Y = Z$, *o`u* $Y,Z \in N$.

*La classe des* langages d'arbres de d´ependances r´eversibles *est la classe g´en´er´ee par les LDG r´eversibles.*

**Théorème 28**
*La classe des langages d'arbres de d´ependances r´eversibles est identifiable.*

*Démonstration.* Nous adaptons l'algorithme d´ecrit en 3.3 de mani`ere `a prendre en compte l'ordre des nœuds et `a consid´erer les symboles d'arit´e multiple.

□

# 6 LES GRAMMAIRES CATÉGORIELLES CLASSIQUES

Bar-Hillel a d´efi ni les *grammaires cat´egorielles*, classe largement utilis´ee dans le domaine linguistique. Les grammaires cat´egorielles sont une logique non-commutative, fragment de calcul de Lambek, lui-même fragment de la logique lin´eaire. Les *types* sont d´efi nis `a partir d'un ensemble de types atomiques par : un type atomique de $Atoms$ est un type et si $A$ et $B$ sont des types, alors $A/B$ et $A\backslash B$ sont des types. Une grammaire cat´egorielle est un quadruplet $\langle \Sigma, Atoms, Lex, S\rangle$, o`u $\Sigma$ est un alphabet et $Lex$ est une fonction qui associe chaque lettre de $\Sigma$ `a un ensemble de types. $S$ est le type fi nal.

Un mot $u_1, \ldots, u_n$ est de type $A$ si il existe un type $T_i \in Lex(u_i)$ pour chaque lettre $u_i$ tel que $T_1 \ldots T_n \rightarrow A$ par applications des r`egles suivantes : $A(A\backslash B) \rightarrow B$ et $(A/B)B \rightarrow A$. Un mot est g´en´er´e par une grammaire si et seulement si il est du type $S$.

**Remarque 29**
*Nous avons vu que les r`egles de productions produites par $\backslash$ et $/$ sont r´eversibles. Plusieurs auteurs ont ´etendu les grammaires cat´egorielles classiques par ajout de nouvelles r`egles. Ces nouvelles r`egles comme la composition produisent elles aussi des r`egles r´eversibles et donc, ces nouvelles grammaires sont elles aussi identifi ables.*

Les *arbres de dérivation* sont d´efi nis comme suit.

- Une lettre $u$ de $\Sigma$ est un arbre de d´erivation ´etiquet´e par un type $T \in Lex(u)$.
- Si $t_1$ et $t_2$ sont deux arbres de d´erivation ´etiquet´es respectivement par $A$ et $A\backslash B$, alors $\backslash(t_1,t_2)$ est un arbre de d´erivation ´etiquet´e par $B$.
- Si $t_1$ et $t_2$ sont deux arbres de d´erivation ´etiquet´es respectivement par $A/B$ et $B$, alors $\backslash(t_1,t_2)$ est un arbre de d´erivation ´etiquet´e par $A$.

Un arbre de d´erivation d´es´etiquet´e est un arbre de d´erivation auquel les ´etiquettes ont ´et´e retir´ees. Dans le livre (Kanazawa, 1998), Kanasawa a ´etudi´e les grammaires rigides. Ces grammaires sont des grammaires cat´egorielles classiques pour lesquelles, `a chaque lettre correspond un et un seul type. C'est `a dire que $Lex$ est une fonction de $\Sigma$ dans l'ensemble des types.

**Théorème 30 (Kanasawa)**
*La classe des grammaires cat´egorielles rigides est identifi able `a partir des arbres de d´erivation d´es´etiquet´es.*

*Démonstration.* L'ensemble des arbres de d´erivation d´es´etiquet´es est un langage r´egulier d'arbres. Pour s'en rendre compte, construisons une GRA normale $\Gamma$.À chaque sous-type $A$ d'un type de lettre, correspond un ´etat $[A]$.À chaque lettre $u$ de $\Sigma$, correspond une r`egle $[Lex(u)] \underset{\Gamma}{\rightarrow} u$. À chaque sous-type $A$ d'un type de lettre, correspond une r`egle : $[B] \underset{\Gamma}{\rightarrow} \backslash([A],[A\backslash B])$ ou $[A] \underset{\Gamma}{\rightarrow} /([A/B],[B])$. On

remarque que ces r`egles sont r´eversibles et puisque la grammaire est rigide, Γ est r´eversible.

Nous concluons par le th´eor`eme 7.

□

# RÉFÉRENCES


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

BIERMANN A. & FELDMAN J. (1972). On the synthesis of finite state machines from samples of their behavior. *IEEE Trans. on Computers*, **21**, 592–597.

COMON H., DAUCHET M., GILLERON R., JACQUEMARD F., LUGIEZ D., TISON S. & TOMMASI M. (1997). Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata.

DIKOVSKY A. & MODINA L. (2000). Dependencies on the other side of the curtain. *Traitement automatique des langues*, **41**(1), 67–96.

EDWARDS J. J., GONZALEZ R. C. & THOMASON M. G. (1976). An algorithm for inference of tree grammars. *International Journal of Computer and Information Sciences*, **5**(2), 145–164.

FERNAU H. (2001). Learning xml grammars. *Machine Learning and Data Mining in Pattern Recognition*, p. 73–87.

FUKUDA H. & KAMATA K. (1984). Inference of tree automata from sample set of trees. *International Journal of Computer and Information Sciences*, **13**(3), 177–196.

GAIFMAN H. (1965). Dependency systems and phrase structure systems. *Information and Control*, **8**(3), 304–337.

HAYS D. (1961). Grouping and dependency theories. In *National symp. on machine translation*.

KAMATA K. (1988). Inference methods for tree automata from sample set of trees. *IEEE International Conference on Systems, Man and Cybernetics*, p. 490–493.

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

KNUUTILA T. & STEINBY M. (1994). The inference of tree languages from finite samples: an algebraic approach. *Theorical Computer Science*, **129**, 337–367.

LEE L. (1996). *Learning of Context-Free Languages: A survey of the litterature*. Rapport interne, Harvard University.

LEVINE B. (1981). Derivatives of tree sets with applications to grammatical inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **3**(3), 285–293.

MÄKINEN E. (1997). *Inferring regular languages by merging nonterminals*. Rapport interne, University of Tampere.

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

SAKAKIBARA Y. (1992). Efficient learning of context free grammars from positive structural examples. *Information and Computation*, **97**, 23–60.

SAKAKIBARA Y. (1997). Recent advances of grammatical inference. *Information and Computation*, **185**, 15–45.