Acquisition de Ressources Linguistiques (S9IN240)
Université de Nantes

Partie I. Méthodes symboliques
Alexandre DIKOVSKY

Cours 2. Acquisition sans tuteur selon Gold. Conditions d’a renabilité.

1 Ensembles révélateurs

Plus bas, nous allons choisir une famille indexée de ressources $\mathcal{L} = \{L(I) \mid I \in \mathcal{I}\}$ et nous allons étudier les conditions d’apprenabilité de $\mathcal{L}$ à la limite sans tuteur (*P-apprenabilité*).

Remarque. La particularité essentielle de l’acquisition sans tuteur est qu’on ne dispose que des exemples positifs $s_1, s_2, s_3, \ldots$ qui tous ensemble représentent une énumération de la ressource cible $L(I_c) = \{s_1, s_2, s_3, \ldots\}$. Alors, on ne sait pas distinguer entre $L(I_c)$ et ses super-ensembles $L(I_c) \subsetneq L(J) \in \mathcal{L}$ sans des conditions additionnelles préalables sur la famille $\mathcal{L}$. Ceci est dû au fait que les exemples pour $L(I_c)$ le sont aussi pour $L(J)$. Alors, il nous faut étudier les conditions suffisantes de P-apprenabilité.

Séquences de verrouillage [M.Blum, L.Blum. I&C, 1975, 28] Soit $I_c \in \mathcal{I}$, une énumeration $\sigma = \{s_1, s_2, ...\} = L(I_c)$ et une fonction $\phi : \{\sigma[n] \mid n \in N\} \rightarrow \mathcal{I}$ telle que

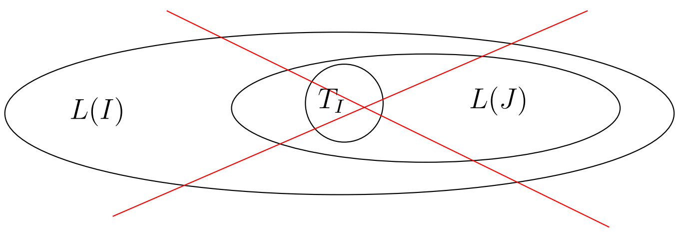
$$\forall n \exists j (\phi(\sigma[n]) \neq \phi(\sigma[n+j])).$$

Alors ϕ n’est pas stabilisée sur σ et donc n’apprend pas $L(I_c)$ sur σ. D’où :

Proposition 1. *Si ϕ apprend $\mathcal{L}$, alors pour tout $I_c \in \mathcal{I}$ il existe une séquence $\sigma_v = \{a_1, ..., a_m\} \subseteq L(I_c)$ (*séquence de verrouillage*) telle que $\phi(\sigma_v) = \phi(\sigma'_v)$ quelle que soit son extension $\sigma'_v = \{a_1, ..., a_m, b_1, ..., b_n\} \subseteq L(I_c)$.*

Ensembles révélateurs

Définition 1 (D. Angluin. I&C, 1980, 45(2)). *Soit un ensemble $L(I) \in \mathcal{L}$. Son sous-ensemble fini non vide $T_I \subseteq L(I)$ est* révélateur *pour $L(I)$ dans $\mathcal{L}$ si $T_I \not\subseteq L(J)$ pour tout sous-ensemble $L(J) \subsetneq L(I)$ appartenant à $\mathcal{L}$.*



Evidemment, si T_I est révélateur pour $L(I)$ alors tout T fini tel que $T_I \subsetneq T \subset L(I)$ est aussi révélateur pour $L(I)$.

Proposition 2. *Soit une fonction ϕ qui apprend $\mathcal{L}$. Si $\sigma_v = \{a_1, ..., a_m\} \subseteq L(I) \in \mathcal{L}$ est une séquence de verrouillage pour $L(I)$ alors σ_v est révélateur pour $L(I)$.*

Démonstration. Supposons que $\sigma_v \subseteq L(J) \subsetneq L(I)$ pour un $L(J) \in \mathcal{I}$. Alors $L(J)$ ne peut pas être apprenable par ϕ. Effectivement, si on prend une énumération $\sigma = \{a_1, ..., a_m, b_1, ...\} = L(J)$ qui commence par σ_v, alors $L(\phi(\sigma[n])) = L(I) \neq L(J)$ pour tout $n \geq m$.
□

Supposons que la famille $\mathcal{L}$ soit récursive. Dans ce cas nous pouvons disposer de plusieurs procédures d'énumération :

Exemple 1. *Décrivez les procédures :*
1. $enum(I, i)$ d'énumération de $L(I)$ (la procédure renvoie l'élément numéro i de $L(I)$ pour toit i).
2. $SecFin(i)$ d'énumération de tous sous-ensembles finis $S \subseteq D$.
3. $SecFinRept(i)$ d'énumération de tous sous-ensembles finis $S \subseteq D$, où tout sous-ensemble est répété un nombre infini de fois.
4. $SecFin(I, i)$ d'énumération de tous sous-ensembles finis $S \subseteq L(I)$.
5. $SecFinRept(I, i)$ d'énumération de tous sous-ensembles finis $S \subseteq L(I)$, où tout sous-ensemble est répété un nombre infini de fois.

Définition 2. *Une famille indexée récursive de ressources $\mathcal{L} = \{L(I) \mid I \in \mathcal{I}\}$ satisfait la* **condition de sous-ensemble révélateur** *s'il existe une fonction récursive (= un algorithme) $ER(I, i)$ qui pour tout $I \in \mathcal{I}$ énumère un* **sous-ensemble (fini) révélateur** *$T_I \subseteq L(I)$ de $L(I)$.*

Théorème 1 (D. Angluin). *Une famille indexée récursive de ressources $\mathcal{L} = \{L(I) \mid I \in \mathcal{I}\}$ est P-apprenable ssi $\mathcal{L}$ satisfait la condition de sous-ensemble révélateur.*

Démonstration. ($\Rightarrow$) Supposons qu'il existe une fonction ϕ qui apprend $\mathcal{L}$ à la limite. Dans ce cas l'algorithme suivant énumère un ensemble de verrouillage de $L(I)$:

EnumVerrou(I, n)
  $a_0 \in L(I)$
  $i \leftarrow 1$
  $j \leftarrow 1$
  $S \leftarrow SecFinRept(I, 1)$
  tant que (**vrai**) **faire**
    si $\phi(S) = \phi(S \bullet SecFinRept(I, j))$ **alors**
      $j \leftarrow j + 1$
      $S \leftarrow S \bullet \{a_0\}$
    sinon
      $j \leftarrow 1$
      $i \leftarrow i + 1$
      $S \leftarrow SecFinRept(I, i)$
    fin si
    si S contient plus de n éléments **alors**
      sortir et renvoyer l'élément numéro n de S
    fin si
  fin tant que

Effectivement, car un ensemble de vérrouillage existe, **EnumVerrou**(I, n) va finalement trouver son numéro i_0 dans l'énumération $SecFinRept(I, i)$. A partir de ce moment i_0 ne change plus et l'algorithme énumère l'ensemble fini de vérrouillage $SecFinRept(I, i_0) \cup \{a_0\}$. Il reste de remarquer que selon la proposition 2 cet ensemble est révélateur pour $L(I)$.
($\Leftarrow$) Supposons qu'il existe un algorithme **EnumRevel**(I, n) qui énumère un ensemble révélateur pour tout $L(I), I \in \mathcal{I}$. Dans ce cas on peut en construire un algorithme **Revelateur**(I, n) qui calcule la séquence (**EnumRevel**$(I, 1)$,...,**EnumRevel**(I, n)) pour tout n et tout $L(I), I \in \mathcal{I}$. On peut aussi définir une énumération $Enum(\mathcal{I})$ de $\mathcal{I}$ avec deux operations d'accès : **first** et **next**. Ensuite on peut définir la procédure suivante :

$\phi((a_1, ..., a_n))$
  $I_0 \in \mathcal{I}$
  $I \leftarrow$ **first** $Enum(\mathcal{I})$
  tant que $(a_1, ..., a_n) \nsubseteq L(I)$ **faire**
    $I \leftarrow$ **next** $Enum(\mathcal{I}, I)$
  fin tant que
  si Revelateur$(I, n) \subseteq (a_1, ..., a_n)$ **alors**
    renvoyer I
  sinon
    renvoyer I_0
  fin si

Lemme 1. *ϕ apprend $\mathcal{L}$ à la limite.*

Démonstration. Soit une séquence d'apprentissage $\sigma = (a_1, ..., a_n, ...) = L(I_c)$ pour un élément cible $L(I_c)$ de $\mathcal{L}$. Quand l'algorithme ϕ trouve dans la boucle le premier index I dans l'énumération **next** $EnumRept(\mathcal{I}, I)$ tel que $(a_1, ..., a_n) \subseteq L(I)$, il effectue le test **Revelateur**$(I, n) \subseteq (a_1, ..., a_n)$. Il y a deux possibilités :
1. le test est positif. Dans ce cas **Revelateur**$(I, n) \subseteq (a_1, ..., a_n) \subseteq L(I) \cap L(I_c)$, c'est-à-dire l'index renvoyé I ne contredirait pas à l'égalté $L(I) = L(I_c)$ si elle avait lieu.
2. le test échoue. C'est possible dans deux situations :
(i) $L(I) \subsetneq L(I_c)$. Dans cette situation l'index renvoyé n'est pas pertinent.
(ii) $L(I) = L(I_c)$ mais n n'est pas suffisamment grand. Dans ce cas l'algorithme omet le bon index I. Par ailleurs, le premier bon index I_h (c'est-à-dire $L(I_h) = L(I_c)$) n'échappe pas à la procédure ϕ pour un n assez grand. Soit le $K \in N$ minimal tel que **Revelateur**$(I_h, K) \subseteq (a_1, ..., a_K)$ et **Revelateur**$(I_h, K) =$ **Revelateur**(I_h). Alors $\phi((a_1, ..., a_n)) = I_h$ pour tout $n \geq I_h$. Donc $\phi((a_1, ..., a_n))$ se stabilise à partir de ce K. □

Corollaire 1. *La famille indexée FIN de tous les langages finis est P-apprenable.*

Démonstration. FIN satisfait la condition de sous-ensemble révélateur : pour un ensemble $F_I, I \in FIN$, $T_I =_{df} F_I$. □

Hélas, la majorité des familles indexées bien connues ne sont pas P-apprenables.

Définition 3. *On dit qu'une famille indexée $\mathcal{L}$ a un* point limite *s'il existe une suite infinie d'ensembles*

$$L(I_1) \subsetneq L(I_2) \subsetneq L(I_3) \subsetneq \ldots, L(I_i) \in \mathcal{L} \text{ pour } i \geq 1 \tag{1}$$

telle que

$$L(I_l) = (\bigcup_{i \in N} L(I_i)) \ \in \mathcal{L}.$$

$L(I_l)$ *est* point limite *de* $\mathcal{L}$.

Corollaire 2 (L.Blum, M.Blum. I&C,1975,28).
Si $\mathcal{L}$ a un point limite, alors $\mathcal{L}$ n'est pas P-apprenable.

Démonstration. Supposons l'inverse. Soit un point limite $L(I_l) \in \mathcal{L}$. Selon la condition de sous-ensemble révélateur, il existe un ensemble fini révélateur $T_{I_l} \subseteq L(I_l)$ qui satisfait cette condition. Selon (1) il y a aussi un $L(I_{i_0}) \in \mathcal{L}$ tel que $T_{I_l} \subseteq L(I_{i_0})$, tandis que $T_{I_l} \not\subseteq L(I_{i_0})$ par définition de révélateur. Contradiction. □

Corollaire 3 (E. Gold. I&C,1967,10).
Si $\mathcal{L}$ est une famille de langages qui contient au moins un langage infini et tous les langages finis, alors $\mathcal{L}$ n'est pas P-apprenable.

Démonstration. $\mathcal{L}$ a un point limite. □

Ce fait implique des conséquences très pessimistes vis-à-vis la P-apprenabilité à la Gold.

Corollaire 4 (E. Gold. I&C,1967,10). *Aucune des familles suivantes n'est P-apprenable :*
- *langages rationnels,*
- *langages linéaires,*
- *langages hors contexte,*
- *langages hors contexte déterministes,*
- *langages contextuels,*
- *langages TAG,*
- *langages récursifs, etc. etc.*

2 Points d'accumulation

Définition 4 (Shyam Kapur. Computational learning of languages. PhD Thesis, Cornell University, Ithaka, New-York, 1991). *On dit que $L(I_a) \in \mathcal{L}$ est un* point d'accumulation *de $\mathcal{L}$ s'il existe une suite infinie d'ensembles*

$$S_1 \subsetneq S_2 \subsetneq S_3 \subsetneq \dots, \tag{2}$$

telle que :
(i) $L(I_a) = \bigcup_{i \in N} S_i$,
(ii) pour tout i il existe $L(I_i) \in \mathcal{L}$ tel que $S_i \subseteq L(I_i) \subsetneq L$.

Remarque 1. *1. Sans perte de généralité on peut supposer que les ensembles S_i dans (2) sont tous finis. Il suffit de considérer une énumération de $D = \{d_1, ..., d_n, ...\}$, ses préfixes $D[n] = \{d_1, ..., d_n\}$ et les intersections $S'_n =_{df} S_n \cap D[n], n \geq 1$. Prouvez que cette suite des ensembles finis vérifie la définition 4.*
2. Tout point limite de $\mathcal{L}$ est aussi son point d'accumulation ($S_i = L(I_i)$).

Théorème 2 (Kapur, ibid). *Soit une famille indexée de ressources* $\mathcal{L} = \{L(I) \mid I \in \mathcal{I}\}$. $L \in \mathcal{L}$ *a un ensemble révélateur ssi* L *n'est pas un point d'accumulation de* $\mathcal{L}$

Démonstration. ($\Rightarrow$) Soit un ensemble révélateur T_I pour un $L(I) \in \mathcal{L}$ qui est un point d'accumulation. Selon la définition 4 $L(I) = \bigcup\limits_{n \in N} S_i$. T_I étant fini, il existe i_0 tel que

$$T_I \subseteq S_{i_0} \subseteq L(I_{i_0}) \subsetneq L(I), L(I_{i_0}) \in \mathcal{L}$$

ce qui contredit au fait que T_I est révélateur pour $L(I)$.
($\Leftarrow$) Supposons que $L(I)$ n'a pas d'ensemble révélateur. Alors $L(I) = \{a_1, ..., a_n, ...\}$ est infini. Soit $S_i = \{a_1, ..., a_i\}$. $L(I) = \bigcup\limits_{i \in N} S_i$. S'il n'y a pas d'ensemble révélateur, alors quel que soit $E \subsetneq L(I)$ fini, il existe un $L(I_E) \in \mathcal{L}$ tel que $E \subseteq L(I_E) \subsetneq L(I)$. C'est vrai pour tous S_i. Alors, $L(I)$ est un point d'accumulation de $\mathcal{L}$. □

Corollaire 5. *Si une famille indexée récursive de ressources* $\mathcal{L}$ *est apprenable alors elle n'a pas de points d'accumulation.*

Angluin et Wright ont trouvé plusieurs conditions suffisantes de P-apprenabilité. Parmi elles les plus utiles en pratique sont les conditions d'épaisseur finie et d'élasticité finie.

3 E aisseur finie

Définition 5. *Soit une famille indexée récursive de ressources* $\mathcal{L} = \{L(I) \mid I \in \mathcal{I}\}$. $\mathcal{L}$ *a une* épaisseur finie *(*finite thickness, *angl.) si pour toute séquence finie d'exemples* $\emptyset \subsetneq S \subseteq D$ *il n'y a qu'un ensemble* **fini** *de ressources* $R(S) = \{L(I) \in \mathcal{L} \mid S \subseteq L(I)\}$ *qui contiennent tous ces exemples.*

Théorème 3 (D. Angluin. I&C, 1980, 45(2)).
Si la famille $\mathcal{L}$ *a une épaisseur finie, alors elle est P-apprenable.*

Démonstration. $\mathcal{L}$ étant récursive, nous pouvons supposer que :
1. pour toute procédure récursive $p(J, i, t)$, où $J \in \mathcal{I}, i \in N$ et t est un nombre qui borne le temps de calcul, il existe un algorithme A_p qui énumère $\mathcal{I} \times N$ **sans répétition** et effectue les appels $p(J, i, t)$ au cours de cette énumération pendant les périodes de temps croissantes : $t = 1$, $t = 2, ...$ Nous allons dire que ce calcul A_p est fait *en parallèle.*
2. il existe un algorithme $Enum(I, n)$ qui calcule $L(I) \cap \{a_1, a_2, \ldots, a_n\}$ où $\{a_1, a_2, \ldots, a_n\} = D[n]$ est la séquence des premiers n éléments dans une énumération standard de D.
Soit l'algorithme suivant :

$RevelEF(I, n)$
  $t \leftarrow 0$
  $a_0 \in L(I)$
  $R \leftarrow (a_0)$
  en parallèle pour tous $J \in \mathcal{I}, i \in N$ **faire**
    si $R \subseteq L(J) \ \wedge \ Enum(J, i) \subsetneq Enum(I, i)$ et le temps des ces calculs est borné par t
    alors
      $R \leftarrow Enum(I, i)$
    sinon

$R \leftarrow R \bullet \{a_0\}$

fin si

$t \leftarrow t + 1$

si R contient plus de n éléments **alors**

sortir et renvoyer l'élément numéro n de R

fin si

fin en parallèle pour tous

Lemme 2. *$RevelEF(I, n)$ énumère un ensemble révélateur de $L(I)$ pour tout $I \in \mathcal{I}$.*

Démonstration. Tout d'abord, selon la définition de $Enum$, si $L(J) \subsetneq L(I)$ alors il existe un i tel que $Enum(J, i) \subsetneq Enum(I, i)$ et un temps t pendant lequel ce calcul est effectué. Par conséquent, si la condition $R \subseteq L(J) \ \wedge \ L(J) \subsetneq L(I)$ est vraie, alors

$$R \subseteq L(J) \ \wedge \ \exists i(Enum(J, i) \subsetneq Enum(I, i)) \qquad (3)$$

est vrai aussi et $RevelEF(I, n)$ trouvera cet i en temps t.

Ensuite, il n'existe qu'un nombre fini de J qui vérifient la condition $R \subseteq L(J) \ \wedge \ \exists i(Enum(J, i) \subsetneq Enum(I, i))$. Effectivement, quel que soit R, $\{a_0\} \subseteq R$. Mais il n'existe qu'un nombre fini de J qui contiennent a_0 parce que $\mathcal{L}$ a une épaisseur finie.

Alors, il existe un nombre maximal $i_{max} \in N$ et le temps maximal t_{max} à partir desquels R ne change plus. Notons cet ensemble fini $R_{max}(I)$. L'algorithme $RevelEF(I, n)$ énumère $R_{max}(I)$. Par définition $R_{max}(I) \subseteq L(I)$. Supposons qu'il existe un ensemble $L(J') \in \mathcal{L}$ tel que $R_{max}(I) \subseteq L(J') \wedge L(J') \subsetneq L(I)$. Dans ce cas la condition (3) est vraie pour un $i' > i_{max}$ tel que $Enum(J', i') \subsetneq Enum(I, i')$, ce qui est calculé en temps $t' > t_{max}$. Par conséquent, $R \leftarrow Enum(I, i')$ et $R_{max}(I) \subsetneq R$. Contradiction. $\square$

Exemple non-trivial d'une famille d'épaisseur finie : langages des patrons

Définition 6. *Soit un alphabet Σ, son sous-alphabet non vide $\Sigma_0 \subseteq \Sigma$ et un ensemble de variables V $(\Sigma \cap V = \emptyset)$. Un patron sur (Σ, Σ_0, V) est un motif $p \in (\Sigma \cup V)^+$. Le langage $L(p)$ du patron p est l'ensemble des motifs $h(p)$, où $h : V \to \Sigma_0^+$ est un homomorphisme sans valeurs vides $(h(x) \neq \varepsilon, x \in V)$.*

Exemple 2. *1. $\Sigma = \{a, b, c\}, \Sigma_0 = \{a, b\}$, $V = \{X\}$ et $p_1 = XcXcX$. Alors $L(p_1) = \{wcwcw \mid w \in \{a, b\}^+\}$.*
2. $\Sigma = \{john, walks\} \cup \Sigma_0$, où Σ_0 consiste des phrases adverbiales anglaises. $V = \{Adv\}$ et $p_2 = john\ walks\ Adv$. Alors $L(p_2)$ contient : ≪john walks far away≫ , ≪john walks fast≫ , etc.

Il est clair que la famille des langages des patrons sur (Σ, Σ_0, V) est une famille indexée récursive de ressources. Outre ça, cette famille a une épaisseur finie. Effectivement, pour tout motif $w \in L(p)$ $|w| \geq |p|$. Ainsi, pour tout motif w il existe un nombre fini de patrons sur (Σ, Σ_0, V) tels que $w \in L(p)$.

Corollaire 6. *La famille des langages des patrons est P-apprenable.*

4 Elasticité finie

Définition 7 (K. Wright. 2d Ann. Workshop on Comput. Learning Theory, Morgan& Kauffman, 1989). *Soit une famille indexée récursive de ressources* $\mathcal{L} = \{L(I) \mid I \in \mathcal{I}\}$. $\mathcal{L}$ *a une* élasticité infinie *s'il existe :*

- une séquence infinie d'éléments $s_0, s_1, s_2, \ldots$ *dans* D

- et une séquence infinie de ressources $L(I_1), L(I_2), L(I_3), \ldots$ *dans* $\mathcal{L}$

telles que pour tout n *:*

$$s_n \notin L(I_n) \tag{4}$$

$$\text{et } \{s_0, \ldots, s_n\} \subseteq L(I_{n+1}). \tag{5}$$

$\mathcal{L}$ *est d'une* élasticité finie *si elle n'est pas d'élasticité infinie.*

Exemple 3. AFD 0-réversibles
Un AFD est 0-réversible *si*
(i) il a un seul état final et
(ii) pour tout état $q \in Q$ *et pour toutes transitions* $\alpha q_1 \to q$, $\beta q_2 \to q$, $\alpha \neq \beta$.

Soit $L_n =_{df} \{w \in (a+b)^* \mid (|w|_a = |w|_b) \ \wedge\ \forall u, v\ (w = uv \to |u|_b + n \geq |u|_a \geq |u|_b)\}$.

Pour tout n, L_n *est accepté par un AFD 0-réversible (c'est-à-dire, est* 0-réversible*).*

Proposition 3 (D.Angluin). *La famille des langages 0-réversibles a une élasticité finie.*

Cette propriété est intéressante grâce au fait suivant.

Proposition 4 (K. Wright, ibid). *Si la famille* $\mathcal{L} = \{L(I) \mid I \in \mathcal{I}\}$ *a une élasticité finie, alors tout* $L(I) \in \mathcal{L}$ *a un ensemble révélateur.*

Démonstration. Si $L(I) \in \mathcal{L}$ n'a pas d'ensemble révélateur, alors

$$\forall T(fini)\ \exists I' \in \mathcal{I}\ (T \subseteq L(I') \ \wedge\ L(I') \subsetneq L(I)) \tag{6}$$

En utilisant (6) nous pouvons définir les séquences (4) et (5) par récurrence :
I. Soit $s_0 \in L(I)$.
II. Soit $s_0, ..., s_n$ et $L(I_0), ..., L(I_n)$ les préfixes des séquences (4) et (5) déjà définis et $s_0, ..., s_n \in L(I)$. Alors selon (6) il y a $L(I') \in \mathcal{L}$ tel que $s_0, ..., s_n \in L(I')$ et $L(I') \subsetneq L(I)$. Il y a un élément $s_{n+1} \in L(I) - L(I')$. Nous pouvons donc définir $I_{n+1} =_{df} I'$. □

Remarque 2. *S'il n'y a pas de contrainte pour que* $\mathcal{L}$ *soit récursive, alors la proposition 4 montre seulement qu'il y a une fonction qui apprh*

Théorème 4 (K. Wright, ibid). *Si une famille indexée récursive de ressources* $\mathcal{L} = \{L(I) \mid I \in \mathcal{I}\}$ *a une élasticité finie, alors elle est apprenable à la limite.*

Démonstration.

Lemme 3. *Si* $\mathcal{L}$ *a l'élasticité finie, alors tout ensemble* $L \in \mathcal{L}$ *a un ensemble de test.*

Lemme 4. *Si dans une famille indexée récursive de ressources* $\mathcal{L}$ *tout ensemble* $L(I) \in \mathcal{L}$ *a un ensemble de test* T_I *alors pour tout* $I \in \mathcal{I}$ *il y a un algorithme qui énumère* T_I.

Nous omettons les preuves. □

Exemple 5 (K. Wright, ibid). *Prouvez que la classe des familles indexées ayant une élasticité finie est fermée par l'union : si* $\mathcal{L} = \{L(I) \mid I \in \mathcal{I}\}$ *et* $\mathcal{M} = \{L(I) \mid I \in \mathcal{M}\}$ *sont deux familles indexées ayant une élasticité finie, alors la famille des unions*

$$\{L(I) \cup M(J) \mid L(I) \in \mathcal{L} \ \wedge \ M(J) \in \mathcal{M}\}$$

a aussi une élasticité finie.

M. Kanazawa [M. Kanazawa, Learnable classes of categorial grammars. 1998, CSLI] a renforcé ce résultat.

Réductions au moyen de relations finiment valuées.

Définition 9. *Soient deux ensembles* E_1 *et* E_2 *et une relation binaire* $R \subseteq E_1 \times E_2$. R *est* finiment valuée *si pour tout* $x \in E_1$ *l'ensemble* $R(\{x\})$ *est fini.*
Pour tout ensemble $L \subseteq E_2$ *:*

$$R^{-1}(L) =_{df} \{x \in E_1 \mid \exists y \in L \ (xRy)\}.$$

*(*image inverse *de* L*).*

Théorème 5 (M. Kanazawa). *Soient deux univers d'objets* $D_{\mathcal{L}}$ *et* $D_{\mathcal{M}}$ *et deux familles indexées de ressources* $\mathcal{L}$ *et* $\mathcal{M}$ *sur ces univers.*

Si $\mathcal{M}$ *a une élasticité finie et* $R \subseteq D_{\mathcal{L}} \times D_{\mathcal{M}}$ *est une relation finiment valuée, alors la famille des images inverses*

$$\mathcal{L} = \{R^{-1}(L) \mid L \in \mathcal{M}\}$$

a, elle aussi, une élasticité finie.

La preuve de ce théorème n'est pas constructive et utilise le lemme de König.

Le théorème 5 est un moyen puissant de preuve de P-apprenabilité des familles des langages.

Exemple 6. *1. Par exemple, le théorème de Kanazawa est vrai pour tous les homomorphismes* R.
2. Soit une famille de langages $\mathcal{L}$ *d'une élasticité finie. Pour un langage* L*, soit* $L^!$ *l'ensemble de toutes les permutations des motifs dans* L*. Alors, la famille des langages*

$$\{L^! \mid L \in \mathcal{M}\}$$

a aussi une élasticité finie.

Corollaire 7. *Soit deux alphabets* Σ_1, Σ_2*. Si* $\mathcal{M}$ *est une famille indexée d'une élasticité finie de langages sur* Σ_2 *et* $R \subseteq \Sigma_1^* \times \Sigma_2^*$ *est une relation finiment valuée, alors la famille des images inverses* $\mathcal{L} = \{R^{-1}(L) \mid L \in \mathcal{M}\}$ *est P-apprenable.*

Corollaire 8. *Si* $\mathcal{M}$ *est une famille indexée récursive d'une élasticité finie de langages, alors la famille* $\{L^! \mid L \in \mathcal{M}\}$ *est P-apprenable.*