Bulletin de l'APMEP n°275-276 - Automne 1970

Fondements mathématiques de la reconnaissance des structures

Monique PAVEL-GIVENS
Maître de Conférences à la Faculté des Sciences de Clermont-Ferrand chargée d'un cours à la Faculté des Sciences de Paris

1. Introduction.

1.1. Définitions du problème.

Reconnaître un objet (image) *I* donné et de catégorie inconnue, c'est associer par son inspection à *I* une des classes (données d'avance) $P_1, P_2, \ldots, P_m$.

1.2. Délimitation du domaine.

La reconnaissance automatique des structures couvre actuellement un domaine d'applications assez bien défini, essentiellement :

a) reconnaissance des caractères écrits (chèques bancaires, cartes de crédit, etc.);

b) reconnaissance de la parole;

c) autres domaines tels que : recherche dans un fichier d'empreintes digitales, tri de plaques photographiques provenant d'expériences de physique nucléaire, analyse photométrique de la cellule sanguine, exploitation des photographies de reconnaissance aérienne, etc.

Cette liste est loin d'être exhaustive; il faut cependant remarquer que presque toutes les applications ont pour but de résoudre l'un ou l'autre de ces deux problèmes voisins que sont la suppression du goulot d'étranglement que constituent les entrées des calculateurs et l'exploitation de documents non codés lorsque leur volume devient intolérable.

Nous nous intéresserons en particulier aux caractères écrits.

1.3. Génération et analyse des structures.

a) *Invariants*

Les structures les plus élémentaires que l'on rencontre usuellement sont les figures géométriques. Elles peuvent être caractérisées par leurs propriétés intrinsèques, i.e. par la constance de certaines fonctions à travers diverses transformations. C'est au reste le but des géométries, définies par la donnée d'un espace topologique et d'un ensemble de transformations de cet espace, que d'étudier les systèmes invariants par rapport à cet ensemble.

Une structure sera donc initialement caractérisée par des invariants — que nous appellerons structuraux — et qui pourront être algébriques ou topologiques.

Lorsque les membres d'une classe diffèrent l'un de l'autre uniquement par une translation de l'axe des coordonnées, une des approches les mieux connues consiste à utiliser les mesures invariantes par rapport aux translations, et la classe la mieux connue de mesures de translations sont les fonctions d'autocorrélation. D'autres fonctions invariantes par rapport aux translations qui ont été utilisées pour caractériser les structures sont les transformées de Fourier et les moments centraux d'ordre supérieur. Malheureusement elles n'ont pas la propriété désirée d'unicité (par exemple, la première fonction d'autocorrélation n'est pas une transformation biunivoque; plusieurs structures qui diffèrent l'une de l'autre par autre chose qu'une translation ont la même fonction d'autocorrélation).

On peut aussi déduire des invariants pour des changements d'échelle et des rotations.

b) Approche descriptive

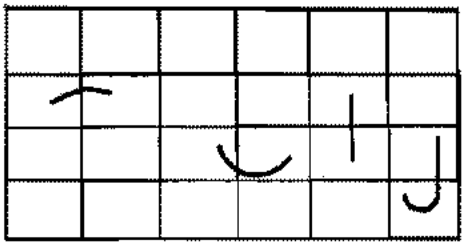
Si les invariants permettent de définir les formes géométriques simples en l'absence de bruit, ou avec une certaine approximation en présence de bruit, ils peuvent devenir inéconomiques dès que celles-ci deviennent complexes (voir, par exemple, les caractères manuscrits de l'alphabet latin, les empreintes digitales, etc.).

On les définit alors descriptivement. Prenons l'exemple de M. EDEN qui décrit les lettres manuscrites de l'alphabet latin à l'aide des quatre éléments ou signes suivants :

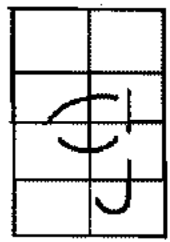
(1) (2) (3) (4)

que l'on combine par concaténation et par un certain nombre de règles précisant l'emplacement des signes au sein d'une lettre et la façon de les joindre (conditions aux limites).

Exemple:



Séquence des signes intervenant dans l'écriture de la lettre *g*.



Représentation de la lettre *g* après l'application des règles de concaténation

On est ainsi conduit à l'idée d'une analyse et donc d'une reconnaissance grammaticale des images. Précisons néanmoins que l'on doit combiner une définition grammaticale de l'ensemble et une définition géométrique de chacun des éléments.

Il se pose un double problème, d'une part de définition grammaticale des structures, d'autre part de reconnaissance des structures, et pour prendre une image commode, d'une part de génération d'un langage, d'autre part de reconnaissance de ce langage.

2. Un modèle mathématique de génération et de reconnaissance des structures.

D'un point de vue formel, le modèle que nous présenterons est linguistique en ce qu'il s'attache à créer un formalisme permettant de décrire et d'analyser une structure, non pas simplement en énumérant des propriétés, mais en mettant l'accent sur la génération des objets que nous observons, i.e. en spécifiant des règles de génération et de transformation par similitude donc en liant les différentes parties d'une image. Cette approche est fondamentalement celle des grammaires génératives.

Les premiers auteurs qui, dans la pratique, ont effectué une analyse grammaticale sont R. A. KIRSCH sur des figures géométriques, R. NARASIMHAN sur des chambres à bulles, LIPKIN-WATT-KIRSCH sur des images biomédicales, M. EDEN sur des caractères manuscrits et GRASSELLI sur des empreintes digitales.

L'idée de concevoir un modèle mathématique du problème en partant de l'analyse grammaticale est due à U. GRENANDER; nous nous en sommes inspirés dans cette étude.

2.1. Exemple : *Un modèle génératif pour les lettres manuscrites anglaises* (Narasimhan et Reddy).

Nous commencerons par un exemple avant de développer la théorie mathématique générale, en espérant qu'il montrera ce que cette théorie se propose de formaliser, dans quelle mesure elle y arrive et quels sont les problèmes de recherche qui restent ouverts.

L'article présenté donne un modèle génératif pour les lettres manuscrites anglaises et la simulation sur calculateur de ce modèle.

Le modèle décrit fait partie d'un modèle plus général, appelé syntactique. Une grammaire à structure de phrase pour un langage L est spécifiée en termes d'un vocabulaire fini V et d'un ensemble fini R de règles de production de réécriture; V est composé de deux parties disjointes $V = \Sigma \cup S$, $\Sigma \cap S = \emptyset$, appelées respectivement vocabulaire terminal et vocabulaire non-terminal; nous appellerons les éléments de Σ mots, ou signes, et ceux de S phrases — ces derniers permettant d'exprimer les règles de R. S contient un élément défini s_0, appelé axiome. Chaque règle de réécriture est de la forme :

$$\alpha A \beta \rightarrow \alpha \gamma \beta$$

où $A \in S$, $\alpha, \beta, \gamma \in V = \Sigma \cup S$. Les règles de réécriture spécifient que dans le contexte $\alpha - \beta$, la phrase A peut être réécrite sous la forme γ. Une grammaire dans laquelle la part du contexte est nulle dans toutes ses règles de réécriture s'appelle grammaire context-free, sinon c'est une grammaire context-sensitive.

On peut donc dire qu'une grammaire (context-free) G pour un langage L est un quadruplet,

$$G = \langle S, \Sigma, R, S_0 \rangle .$$

Chaque règle est une fonction $S \rightarrow S \times \Sigma$, et une règle est terminale si c'est une application $S \rightarrow \Sigma$ (respectivement non-terminale si c'est une application $S \rightarrow S \times \Sigma$).

La grammaire G d'un langage L engendre les propositions de L en partant de s_0 et en appliquant d'une façon répétée les règles de réécriture de R à des phrases appartenant à des chaînes déjà engendrées à ce moment-là. Chaque chaîne de signes qui peut être dérivée de s_0 de cette façon est une proposition de L, et toutes les propositions de L sont dérivables à partir de s_0 de cette façon. Cette procédure de dérivation attribue, d'une manière naturelle, une structure syntaxique à la proposition dérivée. Cette structure syntaxique peut être exhibée sous forme d'un arbre étiqueté.

Exemple

$\Sigma = \{a, b\}$, $S = \{A, B\}$, $s_0 = \{A\}$, $R = \{A \rightarrow aA,\ A \rightarrow bB,\ B \rightarrow bB,\ B \rightarrow b\}$.

On peut alors construire par exemple

$$A \rightarrow aA \rightarrow abB \ldots$$

ou

$$A \to aA \to aaA \to \ldots \to a^m A \to a^m bB \to a^m b^2 B \to$$
$$\to \ldots a^m b^n B \to a^m b^{n+1}$$

et

$$L(G) = \{a^n b^m;\ m \geqslant 0,\ n \geqslant 2\}.$$

Avant de pouvoir appliquer un modèle de structure de phrase de ce genre à un langage bi-dimensionnel, on doit résoudre deux problèmes. On doit redéfinir la notion de « contexte » d'une façon appropriée pour l'appliquer au cas bi-dimensionnel, et la notion de « composition », qui se réduit à la concaténation dans le cas linéaire, doit être généralisée convenablement à deux dimensions. La grammaire donnée ci-dessous est context-free et résout le premier problème d'une façon triviale; la généralisation de la « composition » à deux dimensions implicite dans les règles de réécriture n'est pas triviale.

La figure 1 indique les mots (signes) qui constituent le vocabulaire terminal Σ de la grammaire; les phrases qui constituent le vocabulaire non-terminal S apparaissent dans les membres gauches des règles de réécriture de la *table 1*.

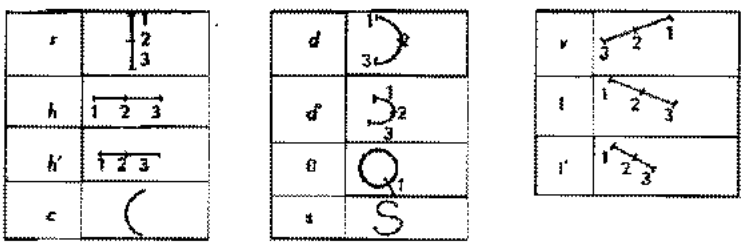


Fig. 1.

Table 1

LETER→|A|B|C|D|E|F|H|...|Z|
A→INVE.*h* (11, 23;)| INVH. *h* (11; 23;)
INVE (1, 2)→*r. l* (11; 2; 2)| *r. v* (11; 2; 2)| *r. l*(11; 2; 2)
INVH (1, 2)→SGMMA. *l* (3, 1; 1; 2)
SGMMA (1, 2, 3)→*r. h'* (11; 2, 3; 3)
B→
⋮
Z→SEVEN. *h* (11;)| SEVEN. *h'* (11;)
SEVEN (1)→*h.r* (31; 3)| *h'.r* (31; ; 3)

Nous ne donnerons pas les définitions (empiriques) générales concernant la table 1; disons seulement que les règles de réécriture doivent spécifier :

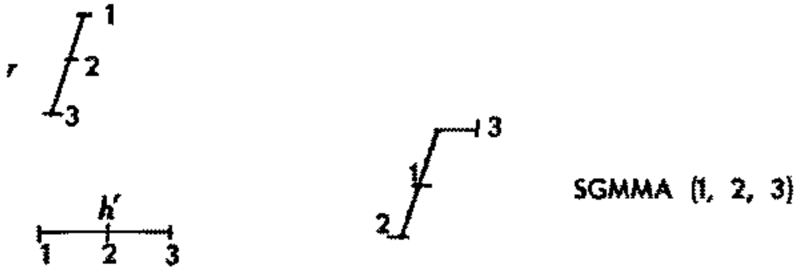
1) quels deux phrases/mots sont composés;

2) les liens des sommets spécifiant la composition;

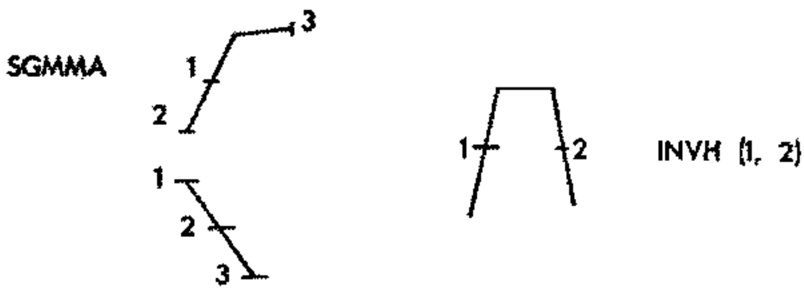
3) la correspondance entre les sommets de la phrase résultante et ceux de ses composantes.

Exemple: un mode de génération de la lettre A.

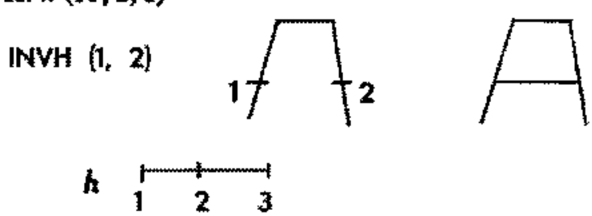
SGMMA $(1, 2, 3) \to r.\, h'\,(11; 2, 3; 3)$



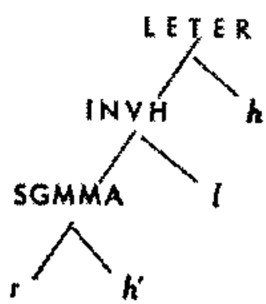
INVH $(1, 2) \to$ SGMMA. $l\,(3, 1; 1; 2)$



A $\to$ INVH. $h\,(11; 2, 3)$



A ce mode de génération de la lettre A correspond l'arbre :



2.2. *Grammaires de structures* (pures).

2.2.1. Signes et configurations.

a) *Signes.*

Les éléments premiers qui permettent de former les configurations, les images et les structures s'appellent *signes*, sont désignés par s, et l'ensemble fini $S = \{s\}$ contiendra le signe vide $\varnothing$ (absence de signes).

Dans ce modèle général, S n'est soumis a priori à aucune condition. Pour nous rendre compte des lois de composition qu'il faut postuler sur S, et aussi pour arriver à classer les éléments de S, prenons l'exemple des caractères majuscules imprimés de l'alphabet latin :

1° On peut considérer des signes composites — éléments de cercles composés avec éléments de droite — et considérer ces derniers comme signes élémentaires;

2° On peut prendre comme signes des éléments de droite et des éléments de cercles;

3° On veut considérer comme semblables des lettres qui ont été obtenues l'une de l'autre par rotation, translation ou dilatation par exemple;

4° Par juxtaposition et superposition de ces signes on peut former les caractères en question.

On postulera donc : une idée d'ordre (partiel) exprimant la différence de complexité des signes; une première notion de classification en différents types; un ensemble de transformations qui conservent cette classification en types; finalement une loi de composition interne sur S, loi qui découlera de celle que nous définirons sur les configurations.

α) Dans la mesure où l'on considère des signes composites, on peut établir une hiérarchie entre les signes élémentaires et les signes composites, c'est-à-dire munir S d'une structure d'ordre partiel. Un signe élémentaire aura pour « complexité » 1 ; un signe composite aura pour complexité le nombre des signes élémentaires dont il est composé.

β) On désignera par Σ l'ensemble des *types* (ou classe) σ_t auxquels peuvent appartenir les signes $s \in S$:

$$s \rightarrow \sigma_t(s) \in \Sigma, \quad \forall s \in S.$$

Le sous-ensemble de S constitué par les signes obtenus à l'aide du type σ_t sera désigné par S_{σ_t} :

$$S_{\sigma_t} = \{s \in S;\ s \in \sigma_t(s),\ \sigma_t \in \Sigma\}.$$

γ) Définissons maintenant une loi de composition externe sur S. On désigne par G un groupe de transformations définies chacune sur S entier :

$$G : S \rightarrow S;$$

Les opérateurs $g \in G$ seront appelés *transformations de similitude*. Deux signes s_1, $s_2 \in S$ seront semblables s'il existe une transformation $g \in G$ telle que $s_1 = gs_2$ et on écrira $s_1 \equiv s_2$ (mod G).

Les hypothèses relatives à la relaxation existant entre la classification en types et la similitude seront exprimées par le

Postulat 1 (sur les types) :

L'ensemble S des signes est divisé en sous-ensembles S_{σ_i}, $\sigma_i \in \Sigma$, les types, $S = \bigcup_i S_{\sigma_i}$, et :

(*i*) $S_{\sigma_i} \cap S_{\sigma_j} = \varnothing$, $\sigma_i \neq \sigma_j$, $\sigma_i, \sigma_j \in \Sigma$;

(*ii*) si $s \in S_{\sigma_i}$, alors $gs \in S_{\sigma_i}$, $\forall s \in S$, $\forall g \in G$.

(*iii*) $g\varnothing = \varnothing$, $\forall g \in G$.

La relation de similitude définie par le groupe G est une relation d'équivalence qui définit une partition sur S :

$$S = \bigcup_{j \in J} S_j, \quad S_i \cap S_j = \varnothing,\ i \neq j; \quad S_j = [s_j] = \{s_p;\ s_p = gs_j\}.$$

On appelle *silhouette* (ou *masque*) un représentant d'une classe d'équivalence.

On a $S/G \subseteq S/\Sigma$.

Problème de la reconnaissance des signes : tout signe $s \in S$, $s \neq \varnothing$, est de la forme $s \rightsquigarrow (t, g)$, où t est une silhouette, et $s = gt$, $g \in G$. Reconnaître le signe s revient à trouver la silhouette t; le problème a une solution unique si g est donné.

b) *Configurations.*

On appelle *configuration* c une suite finie ordonnée de signes :

$$c = (s_1, \ldots, s_n), \text{ avec } s_1, \ldots, s_n \in S,$$

les composantes s_i de c pouvant appartenir à des types S_{σ_i} différents.

On définit sur l'ensemble des configurations une loi de composition interne, notée $+$: si $c = (s_1, \ldots, s_m, s_{m+1}, \ldots, s_n)$, on écrit

$$c = c_1 + c_2, \text{ avec } c_1 = (s_1, \ldots, s_m),\ c_2 = (s_{m+1}, \ldots, s_n),$$

c'est la *concaténation* (cela peut représenter, par exemple, la juxtaposition ou la superposition). Cette loi est *associative*. L'ensemble des configurations est donc un semi-groupe libre avec unité (la suite vide $\varnothing$ est l'élément unité), donc un monoïde libre sur S.

On restreint l'ensemble ainsi défini de toutes les configurations à l'ensemble C des *configurations grammaticales* (ou *significatives*, ou admissibles) en imposant un ensemble R de règles restrictives et des conditions supplémentaires par le

Postulat 2 (sur les configurations grammaticales) :

L'ensemble C des configurations grammaticales, défini à l'aide de R, doit avoir les propriétés suivantes :

(i) si $c = (s_1, \dots, s_n) \in C$, alors $gc = (gs_1, \dots, gs_n) \in C$, $\forall g \in G$.

(ii) si $c = (s_1, \dots, s_n) \in C$, alors $\bar{c} = (s_\alpha, s_{\alpha+1}, \dots, s_\beta) \in C$ pour $1 < \alpha < \beta < n$.

Si toutes les suites finies formées à partir d'un ensemble S de signes s sont grammaticales, nous disons que l'ensemble des configurations est un ensemble de configurations *libres*.

En résumé :

Définition 1.

L'ensemble C des configurations grammaticales c engendrées par l'ensemble S des signes est un monoïde libre engendré par S, à opérateurs de similitude $g \in G$, qui satisfait les trois groupes de propriétés :

I. — 1. $c_1 + c_2 \in C$, $\forall c_1, c_2 \in C$;

2. $(c_1 + c_2) + c_3 = c_1 + (c_2 + c_3)$, $\forall c_1, c_2, c \in C$;

3. $c + \varnothing = \varnothing + c = c$, $\forall c \in C$.

4. $c = (s_1, \dots, s_n) \in C \Rightarrow \bar{c} = (s_\alpha, s_{\alpha+1}, \dots, s_\beta) \in C$, $1 < \alpha < \beta < n$.

II. — G est un groupe défini sur C entier.

III. — 1. $c \in C \Rightarrow gc \in C$, $\forall g \in G$, $\forall c \in C$;

2. $g_1(g_2 c) = (g_1 g_2)c$, $\forall g_1, g_2 \in G$, $\forall c \in C$;

3. $g(c_1 + c_2) = gc_1 + gc_2 \, \forall g \in G$, $\forall c_1, c_2 \in C$.

Définition 2.

On appelle nombre *minimal* de signes d'une configuration c le plus petit nombre naturel n tel que la configuration c puisse être écrite sous la forme $c = (s_1, \dots, s_n)$; ce nombre se désigne par $N(c)$.

Définition 3.

La complexité d'une configuration est la somme des complexités de ses signes.

Définition 4.

Deux configurations sont *identiques* si elles ont le même nombre de signes et si leurs signes sont identiques deux à deux.

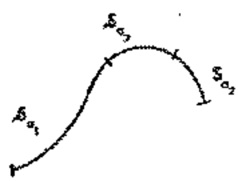
2.2.2. Images.

a) *Définitions et premiers résultats.*

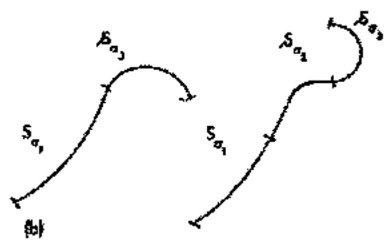
Une image incorpore l'information et la « signification » que lui accorde l'observateur qui peut identifier deux configurations c et c' qui n'ont pas le même nombre de signes et dont les signes ne sont pas identiques; dans ce cas on dira que c_1 et c_2 ne sont pas identiques mais *identifiés* et on écrira $c_1 R c_2$.

Exemple.

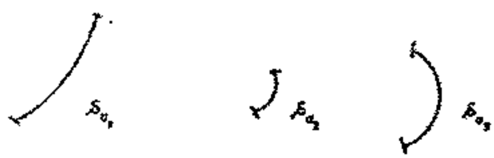
Si l'on considère les classes S_{α_i} consistant en des arcs circulaires et si l'on prend pour G le groupe de transformations euclidiennes du plan, on peut identifier les configurations comme suit. Si $c = (s_1, \ldots, s_n)$, on part de l'origine dans le plan (x, y) et on trace un arc s_1 ayant une extrémité en $x = y = 0$ et ayant sa tangente dans la direction des x positifs. On place à l'autre extrémité l'arc s_2, en lui demandant d'avoir la même tangente que s_1 en ce point particulier. Si l'on prend comme règle R la condition que des arcs voisins sont reliés par des tangentes continues, on obtient une courbe plane consistant en n arcs circulaires. Deux configurations seront identifiées $c_1 R c_2$ si et seulement si les courbes représentatives sont identiques. Par exemple, les configurations (différentes) (a) et (b) doivent être identifiées mais ne sont pas identifiées avec la configuration (c).



(a)



(b)



Postulat 3 (sur l'identification).

La relation d'identification R doit être telle que :

(*i*) Si cRc', alors $gcRgc'$, $\forall g \in G$, $\forall c, c' \in C$

(*ii*) Si $c = c_1 + c_2$, $c' = c'_1 + c'_2$ et $c_i R c'_i (i = 1, 2)$, alors cRc' (R est une congruence sur C);

(*iii*) R est une relation d'équivalence (d'indice fini);

(*iv*) Si c et c' n'ont aucun type en commun, alors ou bien cRc', ou bien $c = c' = \varnothing$.

On démontre alors :

Théorème 1.

Soit C un ensemble de configurations libres et $\mathfrak{I}$ l'ensemble de toutes les classes d'équivalence I engendrées par la relation d'identification R défini sur $C : \mathfrak{I} = C/R$.

On peut alors définir sur $\mathfrak{I}$ une loi de composition externe

$$(g, I) \rightarrow gI, \quad g \in G, \quad I \in \mathfrak{I},$$

et une loi de composition interne

$$(\mathrm{I}_1, \mathrm{I}_2) \rightarrow \mathrm{I}_1 + \mathrm{I}_2, \mathrm{I}_1, \mathrm{I}_2 \in \mathrm{J},$$

telles que :

(i) J soit un monoïde pour la loi interne;

(ii) la propriété de distributivité ait lieu : $g(\mathrm{I}_1 + \mathrm{I}_2) = g\mathrm{I}_1 + g\mathrm{I}_2$;

(iii) la propriété d'associativité ait lieu : $g_1(g_2\mathrm{I}) = (g_1g_2)\mathrm{I}$.

Les classes I seront appelées *images* et l'ensemble J des classes I *une grammaire image*.

Remarquons que l'application d'identification R : C→J n'a pas en général un inverse unique.

En prolongeant la définition de la similitude, des configurations aux images, on obtient la

Définition 5.

Les images I_1, $\mathrm{I}_2 \in \mathrm{J}$ sont *semblables* s'il existe une transformation de similitude $g \in \mathrm{G}$ telle que $\mathrm{I}_1 = g\mathrm{I}_2$.

b) *Opérateur de réduction (ou de simplification). Représentation des images.*

On peut aussi introduire un opérateur de réduction (ou de simplification) μ sur C, en supposant l'existence d'un ordre partiel et une propriété de treillis pour les ensembles de configurations identifiées en une même image. On démontre alors le

Théorème 2.

La condition nécessaire et suffisante pour qu'il existe sur un ensemble C de configurations libres :

a) une relation d'ordre partiel « $<$ », qui permette de comparer deux configurations c et c' appartenant à la même image;

b) une propriété de treillis, en vertu de laquelle tout sous-ensemble $\mathrm{K} \subset \mathrm{C}$ de configurations c appartenant à la même image possède au moins un élément $c_0 \in \mathrm{K}$ tel que $c_0 < c$, $\forall c \in \mathrm{K}$, et

c) un opérateur R qui identifie c et c' si elles appartiennent à la même image, est qu'il existe un opérateur de réduction (ou de simplification) μ de C dans C qui vérifie les conditions suivantes :

(i) $\mu(c) = \mu(c') \Rightarrow \mu(gc) = \mu(gc')$;

$\mu(c_i) = \mu(c_i'), i = 1, 2, \Rightarrow \mu(c_1' + c_2') = \mu(c_1 + c_2)$

(ii) $\mu(\mu(c)) = \mu(c)$;

(iii) $\mu(c) < c$ (*);

(iv) $\mu(c)$ possède au moins un type en commun avec c;

On a alors $c\mathrm{R}c' \Leftrightarrow \mu(c) = \mu(c')$.

(*) On n'utilise pas cette condition lorsque l'on démontre la suffisance de la condition.

c) *Génération et décomposition des ensembles d'images et des images.*

Si une image était décomposée en une somme d'images élémentaires, ce serait un principe de simplification que de vouloir l'identifier en procédant (lorsque c'est possible) composante par composante.

Si on définit, par analogie avec les espaces vectoriels, les notions de dimension et base d'une grammaire image, celle d'indépendance des images et de minimalité d'un ensemble d'images, on est en mesure d'exhiber des conditions nécessaires et suffisantes, ou tout au moins suffisantes, de minimalité des sous-ensembles d'images, et — en ajoutant la condition supplémentaire pour la grammaire image $\mathfrak{I}$ d'être commutative — de décomposition (unique) des images en images types (i.e. en sous-ensembles $I_{k_\sigma} \subset I$, chaque I_{k_σ} étant engendré par les signes d'un certain type σ_k).

2.2.3. Structures (pures)

Comme il est habituel, les structures ne retiendront que les quelques propriétés fondamentales qui suffisent à caractériser un ensemble d'objets, en l'occurrence les images. Deux images différentes peuvent véhiculer la même « signification »; la structure ne veut retenir que cette signification en éliminant la redondance et en ne gardant que des propriétés générales, comme par exemple les différents types qui constituent l'image ou (et) les différentes transformations g qui ont été utilisées pour passer des silhouettes aux signes. On pose alors la

Définition 9.

On entend par *structures* (pures) une famille P de sous-ensembles P_1, P_2, ..., P_n, ... disjoints et stables par G d'une grammaire image $\mathfrak{I}$.

Problème de la reconnaissance des structures (pures) :

Trouver une application (ou fonction de reconnaissance) de $\mathfrak{I}$ dans P qui classe l'image $I \in \mathfrak{I}$ suivant la structure $P_j \in P$ (s'il en existe une) à laquelle elle appartient.

Définition 10.

Soit T un ensemble de transformations t appliquant $\mathfrak{I}$ en lui-même; le *prototype* I_v d'une structure P_v est une image telle que tI_v parcourt une fois et une seule tous les éléments de P_v lorsque t parcourt $T(TI_v = P_v)$.

Le prototype joue, pour la structure, le rôle de la silhouette pour le signe.

T doit contenir le groupe G des transformations de similitude, mais n'est pas obligatoirement un groupe. Lorsque $T = G$ et lorsque l'on peut trouver dans chaque structure un prototype, le classement de l'image I donnée et de structure inconnue consiste simplement en l'analyse unique $I = gI_v$. Dans ce cas particulier et important, on écrit $P = \mathfrak{I}/G$.

Définition 11.

Les images appartenant à une même structure P_ν s'appellent des images *synonymes*. On écrit encore $I_1 \equiv I_2(P_\nu)$.

S'il existe un prototype I_ν, et si T est un groupe, on a $I_1 = t_1 I_\nu$, $I_2 = t_2 I_\nu$, c'est-à-dire $I_1 = tI_2$ avec t_1, t_2, $t \in T$.

Il résulte des définitions 10 et 11 que lorsque $P = \mathcal{I}/G$, c'est-à-dire lorsque $T = G$, la synonymie se réduit à une similitude.

Définition 12.

On appelle *sondage* une fonction f, définie sur $\mathcal{I}$, et telle que si I_1 et I_2 sont synonymes, alors $f(I_1) = f(I_2)$.

Un ensemble $F = \{f\}$ de sondages est appelé *complet* s'il sépare les structures : $f(I_1) = f(I_2)$, $\forall f \in F$, implique que I_1 et I_2 sont synonymes.

Si, enfin, on peut analyser l'image à partir de ses composantes types, c'est-à-dire si l'on peut décomposer I en

$$I = I_1 + I_2 + \ldots$$

et que la reconnaissance de $I_\nu \in P_\nu$ permette d'en déduire la structure de I, on aura *segmenté* le problème.

L'ensemble $\mathcal{G}$ auquel on est arrivé jusqu'à présent :

$$\mathcal{G} = (S, \mathcal{I}, G, P)$$

s'appellera *une grammaire de structures* (*pures*).

2.3. *Grammaires de structures déformées.*

Les images de la vie réelle peuvent rarement être décrites à l'aide d'une simple grammaire de structures pures. Nous considérerons donc que $\mathcal{G}$ représente seulement la première étape de l'analyse et l'appellerons grammaire pure, avec des images et des structures pures. A l'étape suivante nous ajouterons à $\mathcal{G}$ un mécanisme de déformation (bavures, mauvaise impression, etc.), correspondant aux images déformées que l'on rencontre dans la vie réelle.

On a alors le processus de génération suivant :

$$P_\nu \ni I \xrightarrow[D]{} I^D.$$

On a donc d'un côté les images pures qui sont données (elles ont pu être obtenues à la suite à la fois d'observations portant sur une grande quantité d'images réelles — d'où une image moyenne, une photo robot — et à la fois d'un processus d'abstraction) et d'un autre côté des images observables qui sont des images ayant une structure déformée. Les problèmes sont :

1) d'analyser ce mécanisme de déformation,

2) de reconnaître une image déformée (i.e. de lui attribuer une structure évidemment pure, puisque ce sont les structures pures que l'on veut reconnaître), avec le maximum de vraisemblance.

2.3.1. Définitions.

Nous définirons dans ce paragraphe les déformations et leurs rapports avec les lois de composition déjà définies dans l'ensemble des images.

Définition 1.

On appelle *grammaire de structures déformées* (ou *grammaire de déformation*) et on désigne par $\mathcal{G}_{déf}$ le triplet

$$\mathcal{G}_{déf} = (\mathcal{G}, \mathrm{D}, \mathcal{J}^D) \tag{1}$$

où : $\mathcal{G}$ est une grammaire donnée de structures pures $\mathcal{G} = (\mathrm{S}, \mathrm{G}, \mathcal{J}, \mathrm{P})$; D est un domaine donné d'applications d, appelées *déformations*, définies sur $\mathcal{J}$; $\mathcal{J}^D$ est l'ensemble dont les éléments, appelés *images déformées*, sont obtenus en appliquant les déformations $d \in \mathrm{D}$ aux images pures $\mathrm{I} \in \mathcal{J}$:

$$\mathcal{J}^D = (d\mathrm{I};\ d \in \mathrm{D},\ \mathrm{I} \in \mathcal{J});$$

On écrira encore $d\mathcal{J} = \mathcal{J}^D$.

Remarques :

(1) Les déformations peuvent se situer tant au niveau du signe qu'au niveau de la phrase. L'on ne peut pas écrire :

$$d\mathrm{I} = d(s_1 + \ldots + s_n) = \sum_i d_i s_i,$$

mais :

$$d\mathrm{I} = c_0'^D \dot{+} d_1 s_1 + c_1'^D + d_2 s_2 + c_2'^D \dot{+} \ldots \dot{+} d_n s_n + c_n'^D \dot{+} c'^D.$$

Devant l'extrême complexité de cette formulation, on a tendance à donner une définition globale de la déformation de l'image : c'est ce que nous avons fait dans la définition 1. Mais il est évident que par la suite nous serons amenés à simplifier le problème, en supposant en particulier qu'il n'y a pas d'altération de la loi de composition.

(2) d est défini pour un signe $(\mathrm{I} = s)$, mais la connaissance de la valeur de ds pour tous les $d \in \mathrm{D}$, $s \in \mathrm{S}$, ne donne pas, en général, la valeur de $d\mathrm{I}$, puisque nous ne connaissons pas le comportement des déformations d par rapport aux lois de composition des signes.

(3) Si $\mathrm{D} \ni e$, alors $\mathcal{J}^D \in \mathcal{J}$. On désignera dans ce cas par les mêmes symboles d, g, et $+$ les prolongements des applications d, g, et $+$, définis sur $\mathcal{J}$, à $\mathcal{J}^D$. On aura ainsi défini $(\mathrm{I}_1^D, \mathrm{I}_2^D) \mapsto \mathrm{I}_1^D + \mathrm{I}_2^D$, $(g, \mathrm{I}^D) \mapsto g\mathrm{I}^D$, en remarquant toutefois qu'en général $\mathrm{I}_1^D + \mathrm{I}_2^D \neq (\mathrm{I}_1 + \mathrm{I}_2)^D$ et que $g\mathrm{I}^D \neq (g\mathrm{I})^D$. L'élément unité de $\mathcal{J}$ par rapport à $+$ sera toujours l'image E qui contient la suite vide $\{\varnothing\}$.

Définition 2.

On appelle *structures déformées* une famille P^D de sous-ensembles

$$\mathrm{P}_1^D, \mathrm{P}_2^D, \ldots, \mathrm{P}_n^D, \ldots \text{ de } \mathcal{J}^D.$$

Définition 3.

Si D contient la transformation unité, les déformations $d \in D$ s'appellent :

(*i*) *covariantes* si $gd = dg$ a lieu identiquement;

(*ii*) *homomorphes* si $d(I_1+I_2) = dI_1+dI_2$ a lieu identiquement.

Il résulte dans ces conditions que :

1) Lorsque les déformations sont homomorphes, il suffit de les définir pour les signes pour qu'elles soient définies pour les images.

2) Lorsque les déformations d sont covariantes et homomorphes, il suffit de les définir pour les silhouettes pour qu'elles soient définies pour les images.

3) Lorsque les déformations d sont covariantes et homomorphes, elles réalisent des homomorphismes des deux structures algébriques homologues $\mathfrak{J}$ et $\mathfrak{J}^D$.

$\mathfrak{J}^D$ est alors une grammaire image (déformée).

Le problème de la reconnaissance des structures déformées consistera à trouver une application de $\mathfrak{J}^D$ dans P

$$\alpha : \mathfrak{J}^D \to P,$$

appelée *fonction de reconnaissance.*

On s'intéressera aux fonctions de reconnaissance qui classent dans la même structure des images déformées provenant d'images pures synonymes, et des images déformées semblables. En supposant que D contienne la transformation identité, on posera alors la

Définition 4.

Une fonction de reconnaissance $\alpha : \mathfrak{J}^D \to P$ est dite

(*i*) *invariante* (P) si $I_1 \equiv I_2 \pmod{G}$ entraîne

$$\alpha(dI_1) = \alpha(dI_2), \ \forall I_1, I_2 \in \mathfrak{J}, \ \forall d \in D,$$

ou encore si

$$\alpha(dI) = \alpha(dgI) \qquad (2)$$

(*ii*) *invariante* (G) si $I_1^D \equiv I^D \pmod{G}$ entraîne

$$\alpha(I_1^D) = \alpha(I_2^D), \ \forall I_1^D, I_2^D \in \mathfrak{J}^D,$$

ou encore si

$$\alpha(dI) = \alpha(gdI). \qquad (3)$$

On peut alors donner des conditions nécessaires et suffisantes d'équivalence des deux types d'invariance, soit en supposant que D a un élément unité et que les déformations d sont covariantes, soit que D est un groupe contenant G comme sous-groupe normal; dans ce dernier cas, on peut aussi démontrer que la famille de structures $P = \mathfrak{J}/G$ est appliquée par les déformations du groupe quotient D/G dans la famille de structures déformées $P^D = \mathfrak{J}^D/G$.

D'une façon générale, les deux invariances (P) et (G) n'ont pas de raison de coïncider.

2.3.2. Grammaires métriques de déformation.

Nous supposerons, dans ce paragraphe, $\mathcal{I}^D \supset \mathcal{I}$.

Définition 5.

On appelle *grammaire métrique de déformation*, et on désigne par

$$\mathcal{G}_{met} = (\mathcal{G}_{def}, \delta) \qquad (4)$$

une grammaire de déformation $\mathcal{G}_{def} = (\mathcal{G}, D, \mathcal{I}^D)$, où $\mathcal{I}^D$ est métrisé par une distance δ invariante par rapport au groupe G des similitudes.

Cette définition permettra d'introduire des fonctions de reconnaissance G—invariantes.

Nous n'examinerons pas le cas des fonctions de reconnaissance T—invariantes avec $T \neq G$.

a) *Distances invariantes.*

Définition 6.

Une fonction δ, définie sur le produit cartésien $\mathcal{I}^D \times \mathcal{I}^D$ et à valeurs réelles, est une *distance complètement* (resp. simplement) *invariante* si :

(*i*) $\delta(I_1^D, I_2^D) \geqslant 0$; $\delta(I_1^D, I_2^D) = 0$ si et seulement si $I_1^D \equiv I_2^D \pmod{G}$;

(*ii*) δ est symétrique et satisfait l'inégalité du triangle;

(*iii*) $\delta(g_1 I_1^D, g_2 I_2^D) = \delta(I_1^D, I_2^D)$ (invariance complète de δ par rapport à la similitude),

resp. (*iii*)' $\delta(g I_1^D, g I_2^D) = \delta(I_1^D, I_2^D)$ (invariance simple de δ par rapport à la similitude).

Il sera commode quelquefois de supposer que δ prend ses valeurs dans l'intervalle $[0, +\infty]$ de la droite achevée $\overline{\mathbb{R}}$.

Remarques.

1) La condition (*iii*) implique que δ ne dépend que des structures de $P^D = \mathcal{I}^D/G$ qui contiennent les images déformées I_1^D, I_2^D; aussi δ peut être considéré comme étant une fonction $\Delta : P^D \times P^D \to \overline{\mathbb{R}}_+$, $\Delta / \mathcal{I}^D \times \mathcal{I}^D = \delta$.

Ceci n'est pas vrai pour la condition (*iii*)'.

2) Les conditions (*i*) et (*ii*) définissent Δ comme étant une distance.

3) La deuxième partie de la condition (*i*) définit Δ comme étant une fonction qui sépare les structures de P^D : Δ est un sondage dans P^D.

b) *Construction des distances invariantes.*

On démontre que l'on peut construire une distance complètement (resp. simplement) invariante, soit à partir d'une distance quelconque δ_0 définie sur $\mathcal{I}^D \times \mathcal{I}^D$, soit à partir d'une fonction complètement (resp. simplement) invariante d'effort $e : D \to \overline{\mathbb{R}}$ qui satisfait :

(*i*) $e(d) \geqslant 0$; $e(d) = 0$ si et seulement si $d \in G$;
(*ii*) $e(gdh) = e(d)$, $\forall g, h \in G$, *resp.* (*ii*)' $e(g^{-1}dg) = e(d)$;
(*iii*) $e(d_1d_2) \leqslant e(d_1) + e(d_2)$,

si l'on suppose que D est un semi-groupe inversif avec élément unité, contenant comme sous-groupe le groupe G des similitudes.

c) *Construction des fonctions de reconnaissance.*

On déduit des deux types de distances introduites, deux types de fonctions de reconnaissance : les fonctions de reconnaissance à distance complètement invariante minimum, qui pourront être considérées comme des applications de $\mathfrak{I}^D$ dans P, et de P^D dans P, et les fonctions de reconnaissance à distance simplement invariante minimum qui ne pourront être considérées que comme des applications de $\mathfrak{I}^D$ dans P, mais non de P^D dans P.

Définition 7.

Soit δ une distance complètement invariante, définie sur le produit $\mathfrak{I}^D \times \mathfrak{I}^D$ (donc aussi sur $\mathfrak{I} \times \mathfrak{I}$), et soit $I^D \in \mathfrak{I}^D$ une image déformée donnée; on appelle *fonction de reconnaissance à distance minimum* une application

$$\alpha : \mathfrak{I}^D \rightarrow P$$

définie par :

(*i*) α associe à I^D la structure pure $P_\alpha \in P = \mathfrak{I}/G$ pour laquelle $\delta(I, I^D)$ ($I \in P$), est minimum (on suppose que le minimum est atteint pour une seule structure P_α);

(*ii*) si l'on a, pour toutes les images pures, $\delta(I, I^D) = +\infty$, alors l'image I^D ne peut être reconnue.

Remarques.

1) La condition (*ii*) signifie que la déformation d subie par l'image pure a été trop grande : seules les classes de structures déformées qui se trouvent à une distance finie de P peuvent être reconnues. D'une façon générale, ces modèles supposent que l'on a affaire à des petites déformations plutôt qu'à des grandes.

2) Si δ est une distance complètement invariante, alors α minimise la distance $\Delta : P_\alpha \times P_g^D \rightarrow \mathbb{R}_+$, où $P_\alpha \in I$, $P_g^D \in I^D$: α est, dans ce cas, une fonction de reconnaissance G — invariante.

3) On définit d'une façon analogue des fonctions de reconnaissance à effort complètement invariant minimum.

4) On définit, par analogie, des fonctions de reconnaissance à distance (resp. effort) simplement invariante minimum.

2.3.3. Grammaires probabilistes à l'aide des automates.

On peut aussi définir des grammaires probabilistes de déformation et des fonctions de reconnaissance par vraisemblance maximum.

3. Reconnaissance des langages à l'aide des automates.

3.1. Grammaires formelles et automates.

Le but d'une grammaire est de spécifier les règles qui mènent à la construction d'une phrase correcte : on peut parler de grammaires génératives dans le sens qu'elles engendrent des langages. D'autre part, si l'on connaît le langage et on cherche à analyser la phrase, i.e. si l'on s'intéresse à un problème de reconnaissance, on parle d'automates qui sont des mécanismes qui formalisent le processus même de la reconnaissance. CHOMSKY a associé à différentes classes de langages les classes correspondantes d'automates.

Définition 1. Un automate est un quintuplet $a = <S, \Sigma, \delta, s_0, F>$ où

S est l'ensemble fini d'états internes de la machine,

Σ est le vocabulaire fini des entrées,

$s_0 \in S$ est un état initial défini.

$F \subseteq S$ est l'ensemble des états finaux, et sert à délimiter le langage reconnu par l'automate,

$\delta(s, a) = t$ est une application $\delta : S \times \Sigma \to S$, appelée la fonction d'état suivant; si $\delta(s, a) \in F$, alors a est accepté par a.

Exemple.

Nous avons donné la grammaire G définissant le langage :

$$L(G) = \{a^m b^n;\ m \geqslant 0,\ n \geqslant Z\}.$$

Alors l'automate $a = <\{A, B, \varnothing\}, \{\not\subset, a, b, \$\}, \delta, A, \varnothing$,

où $\not\subset$ et \$ sont les extrémités de la bande d'entrée, et

$$\delta(A, \not\subset) = A,\ \delta(A, a) = A,\ \delta(A, b) = B,\ \delta(B, b) = B,\ \delta(B, \$) = \varnothing,$$

accepte le langage L(G) :

$$(A, \widehat{\not\subset}\underbrace{a \dots a}_{m}\underbrace{b \dots b}_{n}\$) \to (A, \not\subset\widehat{a} \dots ab \dots b\$) \to \dots$$

$$\dots(A, \not\subset a \dots \widehat{a}b \dots b\$) \to (A, \not\subset a \dots a\widehat{b} \dots b\$) \to$$

$$\to (B, \not\subset a \dots ab\widehat{b} \dots b\$) \to (B, \not\subset a \dots ab \dots \widehat{b}\$) \to$$

$$\to (B, \not\subset a \dots ab \dots b\widehat{\$}) \to (\varnothing, \not\subset a \dots ab \dots b\$)$$

$$\text{et} \qquad \varnothing \in F.$$

3.2. Problèmes de décision pour les automates finis.

Définition 2. L'ensemble de mots $T(a) = \{x;\ x \in \Sigma^*,\ \delta^*(s_0, x) \in F\}$

est l'ensemble *défini* (ou *accepté*, ou *reconnu*) par a.

Un ensemble $A \subset \Sigma^*$ est appelé *régulier* s'il existe un automate fini a tel que $A = T(a)$.

Les problèmes sont alors les suivants :

1) Caractérisation des ensembles réguliers;

2) Caractérisation de la famille des ensembles réguliers;

3) Finitude des ensembles réguliers;

4) Généralisations possibles.

Mentionnons seulement que tous ces problèmes sont décidables dans le cas des automates finis et c'est là la raison fondamentale pour nous intéresser à cette classe particulière de machines.

Nous avons essayé de mettre rapidement en évidence quelques-uns des problèmes mathématiques liés à la reconnaissance des structures. Notre but principal a été de présenter un modèle algébrique de génération d'images et l'aspect complémentaire de la reconnaissance à l'aide des automates. Il est évident, d'une part, que ce modèle pourrait être étoffé (à l'aide de la théorie des invariants topologiques et algébriques, qui permettrait de formaliser et d'expliciter la nature des transformations de similitude $g \in G$ et des déformations $d \in D$, aussi bien que le seui qui sépare G de D) et rendu « plus » grammatical (en essayant de préciser les règles de production; en tenant compte de la structure grammaticale des images pures lorsque l'on considère les déformations et les fonctions de reconnaissance). D'autre part, le lien existant entre les deux aspects de ce même problème — génération et reconnaissance des images — devrait être approfondi; cela serait possible en faisant une synthèse qui appliquerait à un modèle bien formalisé de génération des images le formalisme mathématique de la théorie des automates (en particulier étude des grammaires bidimensionnelles et automates correspondants).

Disons, enfin, qu'il faudrait considérer le problème, que nous avons supposé résolu, de définition même des classes (décomposition d'un ensemble en classes par rapport à un ensemble de propriétés).

Bibliographie

AHO (A.) et ULLMAN (J.). — The theory of languages. *Math. Syst. theory*, 1968, **2**, n° 2.

GRENANDER (U.). — Foundations of pattern analysis. *IBM technical report*, n° 2, Brown University, Div. of Appl. Math., 1967.

KOLERS (P. A.) et EDEN (M.). — *Recognizing patterns.* MIT Press, 1968.

NARASIMHAN (R.) et REDDY (V.S.N.). — A generative model for handprinted English letters and its computer implementation. *I.C.C. Bull.*, 1967, **6**, n° **4**.

NILSSON (N.). — *Learning machines.* Mc Graw Hill, 1965.

PAVEL (M.). — *Reconnaissance des Structures.* Cours de 3e cycle, chaire de calcul des Probabilités, Faculté des Sciences de Paris. Polycopié, 1967-1968.

PAVEL (M.). — *Fondements mathématiques de la reconnaissance des structures.* Hermann, Actualités Scientifiques, 1970.

RABIN (M. O.) et SCOTT (D.). — *Unite automata and their decision problems. I.B.M. Journ. of Res. and Dev.*, 1959, **3**.

SEBESTYEN (G. S.). — *Decision making processes in pattern recognition.* Mac Millan, 1962.