

Section 2

DÉFINITIONS

Dans un premier temps, nous allons énoncer un certain nombre de définitions ayant trait aux concepts manipulés par les théories de l'information. Nous faisons en sorte que ces définitions et les notations associées soient communes aux différentes théories, afin d'en assurer la cohérence et d'éliminer certaines confusions. Pour éviter toute ambiguïté, nous faisons précéder d'un astérisque les définitions "non-classiques", c'est-à-dire des définitions que nous proposons dans cette étude mais que d'ordinaire les théories n'incluent pas explicitement.

2.1 Les objets

On se donne un *objet* O quelconque : ce peut être un système caractérisé par différents états, un processus défini par un certain nombre de relations, une forme associant plusieurs composants, un langage construit avec des mots, etc. Cet objet doit posséder par hypothèse les trois caractéristiques suivantes :

(i) c'est un objet composite, c'est-à-dire constitué d'un *ensemble*, noté O , de N éléments notés o . N est un nombre supérieur ou égal à 1, ou est infini.

(ii) les éléments o formant O sont clairement *distinguables*, séparables, discernables les uns des autres.

(iii) O est *dénombrable*, i.e. *énumérable*.

***Définition 2.1**

Au sens de la théorie de l'information, un *objet* O est un ensemble dénombrable d'éléments discernables (notés o).

Dénombrer les éléments de O signifie qu'il est possible de les compter, c'est-à-dire d'affecter bijectivement à chaque o un nombre entier ou *index* $i \in \mathbb{N}$ (avec $1 \leq i \leq N$). Cette indexation bijective est cependant effectuée dans un ordre quelconque : tel élément de O sera par exemple indexé par le nombre i au cours d'un premier dénombrement, puis par le nombre j , avec $j \neq i$ en général, à la suite d'un deuxième dénombrement, etc. Bien qu'un dénombrement

s'effectue dans l'ordre des entiers naturels (1, 2, 3, 4,...), cela ne permet pas d'affirmer que l'ensemble O soit muni d'un ordre similaire. Après un dénombrement affectant l'index i à l'élément $p \in O$ et l'index j à l'élément $q \in O$, il vient :

$$i < j \not\Rightarrow p < q$$

car si un tel ordre p existe et *peut* être affecté à O (par un dénombrement), il n'y a aucune raison de supposer qu'il soit défini a priori.

Exemple :

Sur un clavier d'ordinateur considéré en tant qu'objet, je peux compter les touches en suivant l'ordre alphabétique (abcdefghijkl...), l'ordre dactylographique conventionnel (azertyuiopqsd...), l'ordre du code clavier électronique, qui est un code matriciel (touches 1-1, 1-2, 1-3, 2-1, 2-2, 2-3, 3-1...), ou tout autre façon de compter, par exemple de bas en haut et de gauche à droite (=mp:lo;ki,junhy...).

Remarque

Cette définition d'un objet en tant qu'ensemble dénombrable d'éléments discernables est provisoire, puisque nous étendrons par la suite la notion d'objet à celle d'ensemble continu. Notre problématique consistera alors en ceci : à quelles conditions cette extension est-elle possible ? Quelles en sont les conséquences ?

2.2. Les symboles

2.2.1. Du symbole au message

En quoi une telle description des objets peut-elle nous aider à construire le concept de contenu d'information ? Ici, une distinction s'impose. En première approximation, nous considérons les objets comme des entités intangibles, sur lesquelles aucune opération n'est effectuée, ni aucune hypothèse appliquée, autre que les trois hypothèses requises détaillées ci-dessus. Les objets "sont", un point c'est tout... En revanche, parler de "description des objets" suppose un "meta-niveau" distinct du niveau des objets, que l'on a l'habitude de désigner comme étant le monde des symboles. C'est au niveau symbolique que des opérations sont effectuées, que des descriptions existent pour lesquelles l'expression "contenu d'information" pourrait avoir un sens.

Au cours du développement de la théorie, il faudra cependant vérifier la réalité de cette

indépendance de l'information symbolique par rapport au monde des objets. Or on constatera que certains champs de la théorie (théorie quantique notamment) contredisent cette hypothèse simplificatrice.

Définition 2.2

Un *symbole* χ est une forme individuée conventionnelle abstraite élémentaire étiquetant un élément d'un objet.

Forme individuée : un symbole se présente comme un tout distinguable d'un certain environnement.

Forme conventionnelle : au sens d'un processus de communication, le symbole résulte d'un choix arbitraire sur lequel existe un accord préalable entre systèmes communicants. Plus spécifiquement, une condition pour qu'un résultat de la théorie de l'information soit considéré comme valide est qu'il existe un accord sur les conventions de notation (voir plus bas la définition de ce terme), et de symbolisation adoptées (cela s'applique d'ailleurs à n'importe quelle théorie).

Forme abstraite : "abstraire" signifie "isoler par la pensée". La symbolisation est un acte cognitif.

Forme élémentaire : un symbole est par définition "atomique". Décomposer ce symbole en deux morceaux (par exemple couper en deux le symbole mathématique "=") n'a pas de sens. Mais cette remarque nécessite d'être soigneusement précisée : sous l'angle ensembliste, un symbole est un élément insécable par définition. Mais cela ne signifie pas que cet élément ne possède pas une certaine structure interne (cf théorie quantique notamment).

Ces précisions concernant la définition du symbole appellent un commentaire qui a son importance : on constate facilement qu'au niveau le plus bas de la théorie, celui des définitions à partir desquelles il est possible de dériver celle-ci, il est quasi impossible d'ignorer le contenu sémantique, voire subjectif, des notions les plus élémentaires. Notre ambition ne saurait donc atteindre on ne sait quelle objectivité "forte" dans l'élaboration de la théorie de l'information. Nous devons nous contenter d'une intersubjectivité minimale, dont la prétention se limitera en quelque sorte à "objectiver le subjectif" le mieux possible. C'est une des raisons du choix que nous avons fait de désigner cette théorie sous le nom de "théorie multicruciale de l'information", car avant même de savoir comment on pourrait modéliser l'information en respectant son aspect "cognitif", la notion de théorie de l'information suppose d'emblée une part cognitive impossible à évacuer. Mais n'en est-il pas de même de bien d'autres théories,

physiques notamment ?

L'identification des éléments de \mathbf{O} par dénombrement consiste à désigner chacun d'eux par une étiquette, chaque étiquette étant une forme abstraite particulière, c'est-à-dire affecter bijectivement à chaque o un *symbole* χ_i .

Définition 2.3

Une *bibliothèque symbolique* Ξ est un ensemble de symboles, fini ou infini, énumérable et ordonné.

$$\Xi = \{\chi_1, \dots, \chi_i, \dots, \chi_N, \dots\}$$

Cette bibliothèque est telle que, si $p \in \mathbf{O}$ est étiqueté par le symbole χ_i et $q \in \mathbf{O}$ par le symbole χ_j , alors :

$$p \neq q \Leftrightarrow \chi_i \neq \chi_j$$

$$p \equiv q \Leftrightarrow \chi_i \equiv \chi_j$$

Nous supposons donc une stricte bijection entre les éléments de l'objet et leurs symboles. Si cet étiquetage ne préjuge pas de la définition de tel ou tel ordre sur \mathbf{O} , par contre il munit nécessairement la bibliothèque Ξ d'un ordre, en bijection avec l'ordre numérique, fixé par construction. Par exemple, un sac de billes n'a pas d'ordre a priori, mais le fait de compter les billes une à une lui en confère un grâce aux "étiquettes" qui répertorient ces billes, par $\chi_1, \chi_2, \chi_3, \dots, \chi_i, \dots, \chi_N$. Ce premier étiquetage conduit par exemple à affecter à la bille p le symbole χ_i . Un deuxième comptage effectué dans un ordre différent pourrait aboutir à l'étiquetage suivant : $\chi_1, \chi_2, \chi_3, \dots, \chi_j, \dots, \chi_N$. Cette fois la bille p est, par exemple, repérée par le symbole χ_j . Nous dirons que le premier étiquetage construit la bibliothèque Ξ , et que le deuxième étiquetage construit la bibliothèque Ξ' , avec $\Xi' \neq \Xi$.

Exemples :

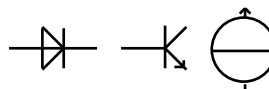
L'acte de symbolisation implique une mise en ordre, quel que soit l'état, ordonné ou non, de l'objet, et indépendamment du ou des ordre(s) qui l'affecte(nt) éventuellement. D'où l'existence de deux cas à considérer :





- Cas des objets non ordonnés : l'objet soumis à symbolisation n'a pas d'ordre a priori.

C'est le cas par exemple de l'ensemble des symboles mathématiques (+, -, ., /, f, (,), x, 1, 2, ...). Hormis les nombres, l'ordre des éléments de cet ensemble est arbitraire. Ou des écritures idéographiques (hiéroglyphes, écritures cunéiformes, chinoises, etc) : à chaque chose, idée, situation, etc, est associé un symbole mais l'ensemble des choses dénotées par ces symboles ne possède pas d'ordre a priori. Il en est de même de l'écriture alphabétique où, en simplifiant, on peut dire qu'à un son est associé une lettre ou un groupe de lettres : bien que l'alphabet soit ordonné (conventionnellement), la liste des sons n'a pas d'ordre a priori.

• Cas des objets ordonnés. Reprenons l'exemple du clavier. Sur cet objet existent par construction et par convention plusieurs ordres : alphabétique, dactylographique, électronique, etc. Supposons que je dispose d'un ensemble de symboles techniques (



 ...) que je veux affecter aux touches du clavier pour réaliser

une bibliothèque de caractères électroniques. L'étiquetage de chaque touche par un symbole électronique est arbitraire (les ordres dont est pourvu le clavier ne me sont ici d'aucune utilité). Par exemple il existe une ressemblance entre la forme du symbole et la forme de la lettre correspondante ( et Z ;  et K, etc). Dans d'autres cas, l'affectation se fait sur la base de la lettre qui code habituellement telle grandeur physique ( et L pour l'inductance, etc). Ou encore tout simplement par ordre alphabétique en fonction du nom du composant ( et T comme "Transistor", etc). Cette opération de symbolisation par un étiquetage ordonné est indépendante des ordres qui préexistent dans le clavier.

• En résumé, l'ordre propre à la bibliothèque est : soit en correspondance avec l'ordre ou l'un des ordres de l'objet, s'il(s) en existe(nt) ; soit arbitraire s'il n'en existe pas. Mais l'affectation d'un symbole à un élément de l'objet est unique. Un changement d'affectation produit une bibliothèque différente.

Définition 2.4

Un *signe* ξ est une variable à valeur dans Ξ : $\xi \in \Xi$.

Sous l'angle statistique, la variable ξ est la réalisation d'un évènement. Généralisant cette

remarque, nous définissons ici le "signe" comme étant "l'actualisation" d'un symbole.

D'une part, cela entraîne que le symbole précède le signe : faute d'une symbolisation préexistante, telle marque, même discernable, ne fera pas signe, et sera ignorée d'un observateur "naïf". Une forme est vide d'information si elle n'a pas été répertoriée préalablement dans une grille de classification, c'est-à-dire une bibliothèque (cf aussi commentaire en définition 2.10).

D'autre part, cela sous-entend que le signe présente un double aspect, statique (i) et dynamique (ii). Soient un ensemble de symboles $\chi_1, \dots, \chi_i, \dots, \chi_N$ et un ensemble de signes $\xi_1, \dots, \xi_k, \dots, \xi_m$. Dire que telle variable (un signe) a telle valeur (un symbole) signifie que :

- (i) $\xi_k = \chi_i$: la variable ξ_k a la valeur χ_i
- (ii) $\xi_k \leftarrow \chi_i$: le symbole χ_i est affecté à la variable ξ_k

Exemples

- Soit un objet technique, comme un amplificateur. Parmi tous les paramètres, notamment toutes les tensions, qui décrivent le fonctionnement de cet amplificateur, la mesure " $u = 10 \text{ V}$ " (de précision supposée limitée à quelques chiffres significatifs) peut être vue comme la réponse à la question "quelle est la valeur de la tension U_k ?" ou comme la réponse à la question "quelle est la tension U_k qui vaut 10 V ?" (i.e. "quelle est la valeur de k pour $u = 10 \text{ V}$?").


- Observation linguistique effectuée sur certaines langues primitives : à la question "qui court ?", c'est-à-dire " ξ est un animal qui court, quelle est la valeur de cette variable ?", les modernes répondent par : "le chien court" (réponse : $\xi = \text{chien}$) mais certains anciens voient le monde autrement et répondraient "la course chienne" (c'est un chien qui réalise l'action de courir).

Définition 2.5

Le *signal* est une matérialisation du signe (sans toutefois préciser la nature de la grandeur physique, qui peut être quelconque : tension, luminosité, etc).

A l'actualisation d'un symbole dans le signe est associée en général la matérialisation du signe dans un support physique ou signal, de sorte que l'on parle parfois du signe comme "symbole physique" (H. Simon).

Définition 2.6

L'inscription ou notation d'un symbole est sa représentation concrète sous forme d'une image géométrique (i.e. une "icône"). On écrira : "tel symbole *noté* χ ", l'inscription  servant à désigner ce symbole.

Il existe une différence de fond entre le monde des objets et le monde des symboles qui les identifient : les premiers sont donnés a priori, les seconds sont arbitraires mais ordonnés, puis peuvent être manipulés par toutes sortes de mécanismes symboliques. Cela suggère l'existence d'une distinction entre les mots "dénombrer" et "énumérer". Ces mots sont considérés comme synonymes au regard de la théorie des ensembles [cf par ex. Wolper 1991, def. 1.7]. Mais ici, dans le cadre d'une théorie de l'information, on pourrait suggérer la distinction suivante :

- un ensemble d'objets O est dénombrable au sens où la mesure de sa taille est évaluée selon sa cardinalité.

- un ensemble de symboles ou bibliothèque Ξ , en plus d'être dénombrable, est énumérable au sens où la mesure de sa taille est évaluée selon son ordinalité.

Rappelons que l'ordinalité n'est définie que pour des ensembles ordonnés, i.e. sur lesquels on a défini une relation d'ordre : deux ensembles ont même ordinalité s'il existe entre eux une bijection qui préserve cette relation d'ordre. Or Ξ est muni d'une relation d'ordre, et il existe une bijection entre Ξ et \mathbb{N} telle que :

$$\chi_i \leq \chi_j \Leftrightarrow i \leq j$$

En théorie des ensembles, les notions de cardinalité et d'ordinalité se confondent pour des ensembles finis, et deviennent distinctes pour des ensembles infinis. En théorie de l'information, cette distinction amène à penser que dénombrer et énumérer sont deux opérations différentes : "dénombrer" consiste à chercher à évaluer le nombre de symboles qui composent la bibliothèque, alors qu' "énumérer" ajoute en plus la notion habituellement sous-entendue de processus d'étiquetage, de désignation bijective ordonnée de chaque composant de l'objet par un symbole numéroté. Un processus de symbolisation est une "mise en ordre" : on rejoint le sens étymologique du mot "ordo" qui, en latin, veut dire file. Alors la mise en ordre par énumération est une opération qui vient logiquement après le comptage par dénombrement.

Or, étant donné un objet et une bibliothèque, tous deux étant des ensembles dénombrables, il y a bien sûr une infinité non dénombrable de façons d'associer bijectivement

des symboles aux éléments de l'objet, car l'ensemble des fonctions de \mathbf{N} dans un \mathbf{N} est un ensemble non-dénombrable. Mais on peut essayer de simplifier cette question en disant que :

- soit il existe dans l'objet, préalablement à toute procédure d'étiquetage, au moins un ordre que l'on peut assimiler à l'ordre numérique.
- soit on peut construire arbitrairement dans cet objet un tel ordre.

Dans ces conditions, un ordre-objet étant donné, les règles de l'étiquetage qui permet de construire une bibliothèque se font plus précises : c'est en ce sens que deux étiquetages effectués selon deux ordres différents construisent deux bibliothèques distinctes. Nous en déduirons les définitions suivantes :

*Définition 2.7

Au sens de la théorie de l'information, est *dénombrable* un objet pour lequel il est possible de compter les éléments dans l'ordre numérique des entiers naturels.

*Définition 2.8

Au sens de la théorie de l'information, est *énumérable* une bibliothèque symbolique pour laquelle il est possible d'ordonner les éléments selon, en référence à, l'ordre d'un ensemble-objet préalablement donné.

Définition 2.9

La $m^{\text{ème}}$ extension d'une bibliothèque Ξ , notée Ξ^m , est le produit cartésien : $\Xi^m = \Xi \times \Xi \times \dots \times \Xi$ (m fois).

Définition 2.10

Un *message* σ est une suite finie ordonnée de signes. Un message de m signes est donc un élément de Ξ^m . On note :

$$\sigma = \xi_1 \xi_2 \dots \xi_k \dots \xi_m \text{ avec } m \in \mathbf{N}, \xi_k \in \Xi \text{ et } \sigma \in \Xi^m.$$

On note $\varepsilon = ""$ le message vide.

En se limitant à une interprétation sémiotique simple, cette définition motive le choix que nous faisons du terme "signe" dans la définition 2.4 : le signe possède d'une manière générale un double aspect, signalétique et sémantique. Cela se traduit ici par le fait qu'un signe

est à la fois une variable symbolique et une variable de position (le signe renseigne sur le fait qu'à tel endroit dans le message on trouve tel symbole). En se contentant d'une acception très pauvre, mais simple à modéliser, de la notion de signification, un symbole χ_i donné n'a pas la même "signification" selon qu'il se situe à la position j ($\xi_j = \chi_i^{(j)}$) ou à la position k ($\xi_k = \chi_i^{(k)}$) dans le message. Cette idée élémentaire sera sous-jacente tout au long de notre analyse.

Définition 2.11

Le dictionnaire Ξ^* est l'ensemble des extensions de Ξ , et donc l'ensemble des messages de longueur finie. C'est la réunion des extensions de Ξ .

$$\Xi^* = \bigcup_{m \geq 1} \Xi^m$$

Théorème 1.1

Ξ^* est énumérable.

- Anticipant sur les définitions 1.18 et suivantes, on peut imaginer la procédure suivante : Ξ étant énumérable, on affecte à chaque symbole de Ξ une chaîne binaire finie (par exemple la transcription en code unaire de son numéro d'indexation). Tout message étant une suite finie de signes, on affecte à chaque signe du message un numéro d'ordre : 1 pour le premier signe, 2 pour le second, etc. On transcrit chaque signe dont le numéro est impair par une suite de "1", et chaque signe de numéro pair par une suite de "0" (ce qui permet de distinguer facilement les signes les uns des autres dans la transcription unaire d'un message). Alors chaque message correspond bijectivement à une suite finie de bits – qui est identifiable à un nombre entier. L'ensemble des messages est représenté par un sous-ensemble de \mathbb{N} et est donc énumérable.



Définition 2.12

Ξ^* est munie d'une fonction de volume $l(\sigma)$ qui est la *longueur* d'un message, exprimée en nombre de signes : Si $\sigma \neq \varepsilon$, $l(\sigma) = m$ sinon $l(\varepsilon) = 0$.

Définition 2.13

Ξ^* est munie d'une relation binaire, la *concaténation*, notée \mathcal{E} , qui associe à toute paire

ordonnée $(\rho, \sigma) \in \Xi^* \times \Xi^*$ l'élément $\rho\sigma = \mathcal{E}(\rho, \sigma)$. Cette opération est associative et a ε pour élément neutre. Ξ^* est un ensemble clos pour cette relation.

2.2.2 Source symbolique

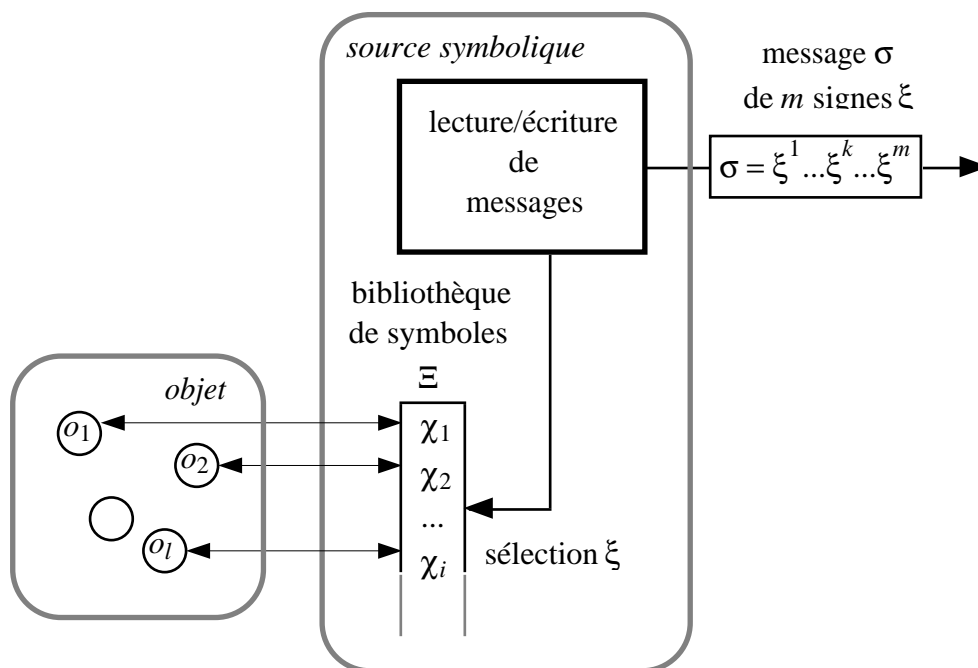
Définition 2.14

Une *source symbolique* Σ est un sous-ensemble de Ξ^* : $\Sigma \subset \Xi^*$

Une source symbolique Σ est définie par :

- a)- une bibliothèque de symboles Ξ
- b)- une ou plusieurs règles de sélection des symboles de Ξ pour lire ou écrire un signe.
- c)- une ou plusieurs règles de lecture/écriture de suites de signes appelées "messages".

Une source symbolique est munie d'un ordre numérique, qui est celui de la bibliothèque.



• Figure 2.1 : source symbolique

Ces définitions appellent quelques commentaires.

- Le terme "source" est pris dans un sens général. Notamment, à deux objets **A** et **B** correspondront respectivement deux sources Γ et Σ , de telle sorte que, si ces deux objets font

partie d'un processus de communication, où A serait le récepteur et B l'émetteur, on parlerait de la "source réceptrice" Γ et de la "source émettrice" Σ , comme en parle en électricité de "source" (de tension ou de courant) génératrice ou réceptrice. C'est pourquoi nous employons l'expression "lecture/écriture" dans le schéma d'une source symbolique.

- Cas particulier où $m = 1$ pour tous les messages : si l'on restreint la définition de Σ à l'ensemble des messages de longueur unité (cas fréquent), on parle alors par abus de langage de "messages" à propos des "symboles", et de "source" à propos de "bibliothèque". Mais cette confusion occulte le processus complémentaire qui consiste à construire une juxtaposition ordonnée de suites de symboles formant message, selon des règles d'écriture et de lecture qui sont évidemment sans objet lorsque la longueur des messages est limitée à l'unité !

- Enfin, on notera que tout ce qui vient d'être dit à propos des symboles a pour corollaire qu'une théorie de l'information est une théorie des symboles : les objets n'apparaissent qu'indirectement dans celle-ci, seuls sont dénombrées, énumérées, manipulées leurs représentations symboliques.

2.3. Les mots

Ce passage du monde des objets au monde des symboles est-il suffisant pour construire une théorie de l'information ? Si oui, cela reviendrait à dire que le procédé idéographique est un procédé général (et suffisant) : on pourrait décrire l'ensemble des individus de nationalité française par l'ensemble de leurs photos d'identité, comme on le fait pour l'ensemble des élèves d'une classe à l'aide du bien connu "trombinoscope"... Or il est clair qu'une feuille contenant les 30 photos d'une classe de 30 élèves est une description certes bijective de "l'objet" qu'elle représente, mais contient une information potentiellement assez pauvre (bien qu'ordonnée : les photos sont disposées sagement de gauche à droite et de haut en bas sur le trombinoscope, alors que la classe elle-même est plutôt un objet désordonné...).

Dans l'analyse du contenu d'information d'un objet intervient alors la notion d'écriture alphabétique.

Définition 2.15

Un *alphabet* A est un ensemble fini énumérable et ordonné de Q symboles ($Q \in \mathbb{N}$). Appelons *lettres* les symboles notés " X " constituant cet alphabet. Alors :

$$A = \{X_1, X_2, \dots, X_i, \dots, X_Q\}.$$

Un alphabet est donc une bibliothèque finie.

Définition 2.16

Q est la cardinalité de A et est encore appelée sa *valence*.

Remarque : par extension, nous appellerons également *valence* la cardinalité d'une bibliothèque finie : $N = \text{card}(\Xi)$.

Définition 2.17

Un *caractère* x est un signe alphabétique, c'est-à-dire une variable à valeur dans A : $x \in A$.

Remarque : une écriture alphabétique est donc un cas particulier de source finie. Réciproquement une source finie constitue en soi, ipso facto, un alphabet. C'est pourquoi l'on parle souvent de "caractère" à propos des symboles d'une source, bien que la notion de caractère soit plus restrictive que la notion de signe : un ensemble de symboles est fini ou infini, un ensemble de caractères est fini par définition. Il peut paraître redondant de spécifier les définitions des sources au sens général du terme, puis des sources alphabétiques, puisque cela amène à des concepts très semblables. Cette démarche a pour but de suivre au plus près la démarche cognitive qui consiste à nommer symboliquement les objets, puis à les transcrire dans un second temps en écriture alphabétique.

Définition 2.18

La $n^{\text{ème}}$ *extension* d'un alphabet A , notée A^n , est le produit cartésien :

$$A^n = A \times A \times \dots \times A \quad (n \text{ fois})$$

Définition 2.19

Un *mot* ou *chaîne*, noté s , est une suite finie de n caractères ($n \in \mathbb{N}$) et est un élément de A^n . On note :

$$s = x_1 x_2 \dots x_j \dots x_n \quad \text{avec } x_j \in S \text{ et } s \in A^n$$

On note $\varepsilon = ""$ le mot vide.

Définition 2.20

Le *vocabulaire*, noté A^* , est l'ensemble des mots de longueur finie écrits sur A . C'est la réunion des extensions de A .

$$A^* = \bigcup_{n \geq 1} A^n$$

Remarque : si les mots sont considérés (par abus de langage) comme des symboles, alors l'ensemble des mots A^* serait une bibliothèque. Mais on va voir que cette interprétation est illicite.

Théorème 1.2

A^* est énumérable.

□ (démonstration similaire au théorème 1.1).

□

Définition 2.21

Un *texte* t est un message particulier formé d'une suite finie de m mots ($m \in \mathbb{N}$). Un texte est donc une suite de suites de caractères. C'est donc aussi un mot ($t \in A^*$).

$$t = s_1 s_2 \dots s_k \dots s_m$$

Définition 2.22

A^* est munie d'une fonction de volume $l(s)$ qui est la *longueur* d'un mot, exprimée en nombre de caractères :

$$\text{Si } s \neq \varepsilon, l(s) = n \quad \text{sinon } l(\varepsilon) = 0.$$

Définition 2.23

A^* est munie d'une relation binaire, la *concaténation*, qui associe à toute paire ordonnée $(r,s) \in A^* \times A^*$ l'élément rs . Cette opération est associative et a ε pour élément neutre. A^* est un ensemble clos pour cette relation.

En particulier, on considèrera l'alphabet binaire $B = \{0,1\}$

On note $b \in B$ un symbole binaire ou *bit*.

On note $B^* = \{ "", "0", "1", "00", "01", "10", "11", "000", \dots \}$ l'ensemble des chaînes de

caractères, ou mots, sur cet alphabet.

Soit w un mot de \mathbf{B}^* .

(i) Il existe une relation bijective \mathcal{B} entre \mathbf{N} et \mathbf{B}^* , par exemple l'écriture binaire des entiers naturels. Pour écrire cette relation, définissons la fonction *quasilogarithme* binaire par :

$$\text{qlog } x = \begin{cases} \log_2 x & \text{si } x > 1 \\ 0 & \text{si } x \leq 1 \end{cases}$$

On note la correspondance entre chaîne et nombre par le signe \cong . Il vient, en notant $\lambda \in \mathbf{N}$ un entier quelconque et w son écriture binaire :

$$\lambda \cong "b_n \dots b_1 b_0" \Leftrightarrow \begin{cases} \lambda = \mathcal{B}^{-1}(w) & : \quad \lambda = \sum_{i=0}^n b_i 2^i \\ w = \mathcal{B}(\lambda) & : \quad \begin{cases} b_0 = \lambda \bmod 2 ; \lambda_1 = \lambda \setminus 2 \\ b_1 = \lambda_1 \bmod 2 ; \lambda_2 = \lambda_1 \setminus 2 \\ \dots \\ b_n = \lambda_n \bmod 2 ; \lambda_n \setminus 2 = 0 \Leftrightarrow n = \lfloor \text{qlog } \lambda \rfloor \end{cases} \end{cases}$$

(remarque : si $\lambda \geq 1$, " b_n " \cong 1)

Ce faisant, l'écriture des symboles au moyen d'un alphabet introduit un nouvel ordre qui vient s'ajouter à l'ordre numérique. Il s'agit de l'ordre lexicographique :

(ii) \mathbf{B}^* est muni d'une relation d'ordre total, l'ordre lexicographique, noté " $<$ ", qui est défini par :

(a) $\forall w \neq \varepsilon, \varepsilon < w$

(b) $\forall v = "x_1 \dots x_p"$ et $w = "y_1 \dots y_q"$, $v < w$ ssi :

(α) v est un préfixe de w : $p < q$ et $x_i = y_i \forall i \in [1, p]$

ou (β) $\exists j \in [1, \min(p, q)]$ tel que : $x_i = y_i \forall i \in [1, j-1]$ et $x_j < y_j$

Par exemple " 10100 " $<$ " 1011 " d'après la règle (β) (en prenant $j = 3$) et " 10100 " $<$ " 101000 " d'après la règle (α).

Remarque : la fonction successeur dans l'ordre lexicographique est indéterminée, car à la chaîne " $y0$ " on peut faire succéder la chaîne " $y1$ " ou la chaîne " $y00$ ".

(iii) Il existe une correspondance bijective \mathcal{L} entre \mathbf{N} et \mathbf{B}^* suivant l'ordre lexicographique :

$(\varepsilon, 0), ("0", 1), ("1", 2), ("00", 3), ("01", 4), ("10", 5), ("11", 6), ("000", 7), ("001", 8), \dots$

On note : pour $\lambda \in \mathbf{N}$ et $w \in \mathbf{B}^*$, $w = \mathcal{L}(\lambda)$.

Si $w \neq \varepsilon$, on pose : $w = "x_1 \dots x_n" = "0^{n-(m+1)} x_m x_{m-1} \dots x_1 x_0"$ pour $m \in [0, n-1]$.

Il vient :

$\lambda = \mathcal{L}^{-1}(w)$:

$$\left. \begin{array}{l} n = l(w) \\ t = \sum_{i=1}^n x_i 2^{n-i} \text{ si } n \geq 1, \quad t = 0 \text{ sinon} \end{array} \right\} \Rightarrow \lambda = 2^n - 1 + t$$

$w = \mathcal{L}(\lambda)$:

La fonction de volume est définie par :

$$l(w) = n = \lfloor \log(\lambda + 1) \rfloor.$$

Il vient :

$$\left. \begin{array}{l} t = \lambda - 2^n + 1 \\ m = \lfloor \text{ql} \log t \rfloor \\ "x_m \dots x_1 x_0" \cong t \end{array} \right\} \Rightarrow w = "0^{n-(m+1)} x_m x_{m-1} \dots x_1 x_0"$$

On lit dans la littérature des affirmations telles que celle-ci : «*It is convenient not to distinguish between the first and second element of the same pair [i.e. une paire $(x \in \mathbf{N}, y \in \mathbf{B}^*)$ dans l'ordre lexicographique], and call them "string" or "number" arbitrarily*» [Li, Vitányi, 1997, p12]. Bien qu'il existe une bijection entre ces deux ensembles, il faudrait nuancer cette affirmation. Il est possible d'énumérer les entiers naturels dans l'ordre numérique croissant à partir de zéro, en affectant à chaque nombre un nom : c'est le principe de la numération en tant que système d'écriture symbolique des nombres. Mais réciproquement, s'il est possible d'énumérer les éléments de \mathbf{B}^* dans l'ordre numérique, cela ne l'est plus dans l'ordre lexicographique, puisque l'on rencontre au cours de l'énumération des suites infinies de mots :

\mathbf{B}^* énuméré dans l'ordre $>$ de \mathbf{N} :

ε

\mathbf{B}^* énuméré dans l'ordre $>$ de \mathbf{B}^* :

ε

0 1 00 01 10 11 000 001 010 011 100 101 110 111 0000 0001 ...∞	0 00 000 ...∞ 001 ...∞ 01 010 ...∞ 011 ...∞ 1 10 100 ...∞ 101 ...∞
--	--

• Tableau 3.3 : le signe "...∞" symbolise une suite infinie de mots.

En ce sens, N est "plus fondamental" que B^* , puisqu'il est nécessaire de connaître l'ordre numérique pour énumérer B^* . Cette distinction apparaît quand il faut par exemple calculer la distance entre deux mots. Un langage formé d'un ensemble de mots de B^* ordonné selon $<$ est décidable par une machine de Turing. Dans le cas le moins favorable, il faut que la machine compte jusqu'à un certain rang maximum pour décider si une chaîne lue est ou non un mot du langage (dans l'exemple cité, la distance entre deux mots est toujours finie). Un tel langage est récursif. Par contre un langage formé d'un ensemble de mots de B^* ordonné selon $<$ conduit à des exécutions infinies (la distance entre deux mots peut être infinie). On retrouve ici la distinction entre dénombrement et énumération évoquée plus haut lorsqu'il s'agit non seulement de compter mais aussi d'étiqueter les éléments d'un objet.

2.4. Les textes

2.4.1. La transcription alphabétique des messages en textes

Définition 2.24

Un langage L est un sous-ensemble de A^* .

Théorème 2.3

L'ensemble des langages n'est pas énumérable.

- Un langage est une partie de A^* . Donc l'ensemble des langages est l'ensemble des parties $\mathcal{P}(A^*)$. On sait qu'un tel ensemble n'est pas énumérable.

□

Définition 2.25

Un *codage* (ou *transcodage*) est une application, notée E , de A^* dans A^* . Un *code* est un langage, image du codage.

Soit $y = E(x)$. Par définition, $x \in L \subseteq A^*$ et $y \in L' \subseteq A^*$. x et y sont appelés respectivement *mot-source* et *mot-code*. On dit encore que y est l'*encodage* de x , également noté $\langle y, x \rangle$. L'opération qui consiste à calculer $x = E^{-1}(y)$ est appelée *décodage*.

Définition 2.26

Un code est dit *régulier* si cette application est injective : il n'y a pas d'homonymes. Un code non régulier est *singulier*.

Mais rien n'empêche qu'un même mot-source soit codé par plusieurs mots-code (synonymie). On parle alors de code *dégénéré*. C'est le cas notamment du code génétique [Tabary, 1987].

Définition 2.27

Un code est *déchiffrable* (i.e. décodable de façon univoque, ou uniquement décodable) si deux textes distincts formés de messages en nombres non nécessairement égaux ont des codages différents.

Par exemple le code ("0",A), ("10",B), ("01",C) est régulier mais non déchiffrable : le message "010" peut se traduire par AB ou CA.

On peut définir l'extension \mathcal{E} du codage E aux suites de mots comme étant l'application de A^* dans $(A^*)^*$, munie de l'opération de concaténation :

$$\mathcal{E}(\varepsilon) = \varepsilon \quad (\text{un texte vide est un mot vide})$$

$$\forall t \in (A^*)^*, \forall y = E(x) \in A^*, \mathcal{E}(t \cdot y) = t \cdot E(x)$$

Définition 2.28

Un *arbre de codage* est la représentation graphique d'un code sous la forme d'un arbre enraciné ordonné d'arité maximale Q .

Rappels : un arbre est un graphe orienté, connexe, acyclique. Le sommet de l'arbre est un nœud particulier nommé *racine* à partir duquel on parcourt l'arbre. Si le dernier arc (i.e. *branche*) sur le chemin de la racine r vers un nœud x est (y,x) , alors y est le *père* de x , et x est le *fil* de y . La racine est un nœud sans père. Un nœud sans fils est un *nœud externe* ou *feuille*. L'arbre est ordonné si les fils d'un nœud sont ordonnés : c'est le cas d'un arbre de codage, où l'arc menant au premier fils code pour la première lettre de l'alphabet, le deuxième pour la deuxième lettre, etc. Le nombre maximal de fils d'un nœud (i.e. arité) est Q , valence de l'alphabet.

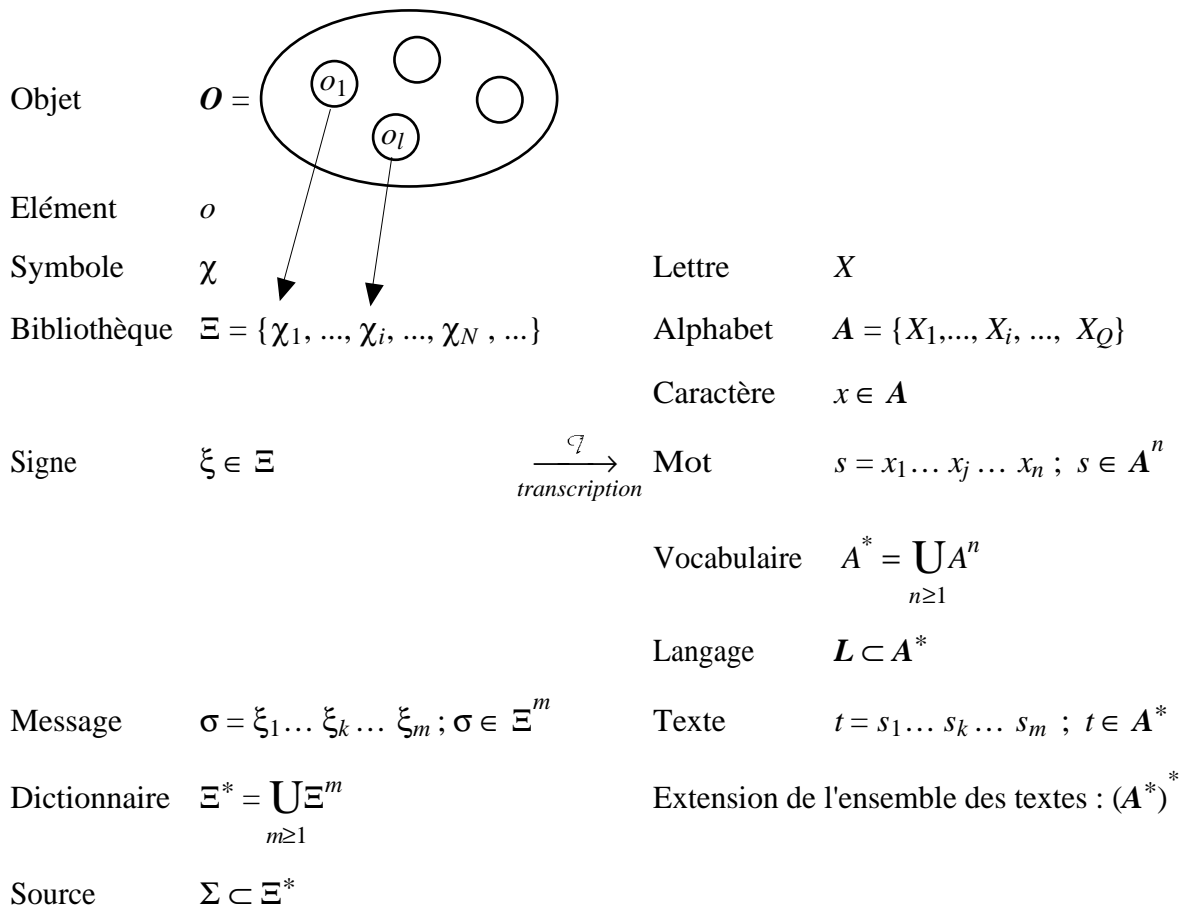
Chaque mot-code w est l'étiquette du nœud x (i.e. l'information attachée à x). Il est constitué de la suite de lettres codée par la suite des branches formant le chemin (r,x) .

**Définition 2.29*

Une *transcription* notée \mathcal{T} est une application de Ξ dans A^* .

Soit $s = \mathcal{T}(\xi)$. Par définition, $\xi \in \Xi$ et $s \in L \subseteq A^*$. Alors s est le mot-code du symbole ξ . Par analogie avec la notion de code, nous dirons qu'une transcription est régulière si un mot-code est la transcription d'au plus un symbole et déchiffrable si la transcription d'un message est décodable de façon univoque. Elle est représentée graphiquement par un arbre de codage.

Arrivés en ce point, nous pouvons rassembler en un tableau les principaux concepts définis dans cette section :



• Tableau 2.1 : résumé des notations introduites dans cette section

2.4.2. Source alphabétique

Nous allons maintenant envisager l'écriture ou la lecture des textes d'un point de vue pratique, en fonction des caractéristiques de la bibliothèque symbolique qu'il faut transcrire (bibliothèque finie ou infinie), des caractéristiques des messages produits par la source (messages de longueur unité ou ou de longueur supérieure à un), de l'alphabet disponible et des caractéristiques de l'arbre de transcription. La réunion de toutes ces ces conditions conduit à de très nombreuses possibilités de lecture/écriture des textes, que nous allons classer en quelques grandes catégories.

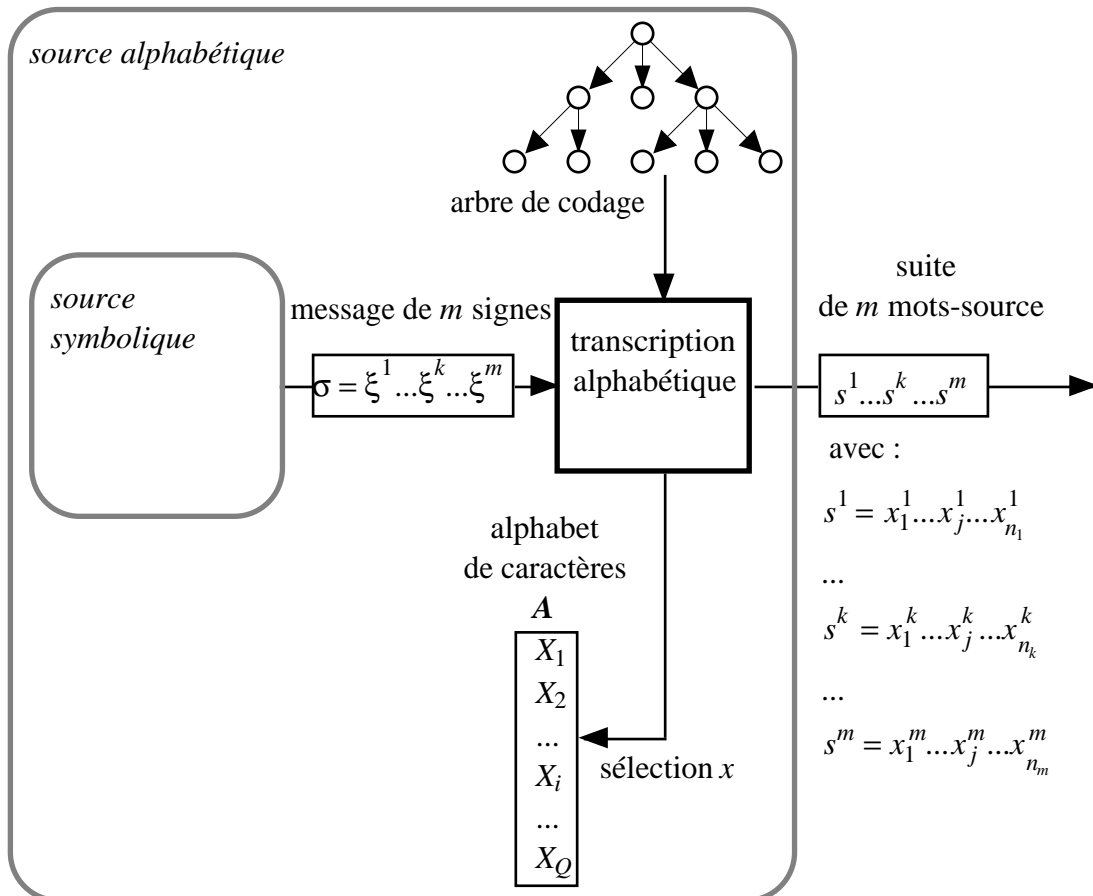
*Définition 2.30

Une *source alphabétique* S est une source dont les symboles sont transcrits alphabétiquement dans un langage.

Une source alphanumérique S est définie par :

- a)- une source symbolique Σ
- b)- un alphabet A
- c)- un arbre de codage pour la transcription alphanumérique des symboles en mots d'un langage.

d'un langage.



• Figure 2.2 : source alphanumérique

Une difficulté apparaît : langage et source ne se situent pas au même niveau dans le tableau 2.x, et ne sont pas du même type. Un symbole étant par définition une entité distinguable, individualisée, l'écriture d'un message sous la forme d'une suite de symboles est par hypothèse déchiffrable. De même un mot est constitué d'une suite de caractères que l'on suppose implicitement distinguables les uns des autres dans le mot, car l'alphabet est lui-même une bibliothèque de symboles (les lettres) eux aussi distinguables par définition. Par contre la transcription d'un message sous la forme d'un texte ne sera déchiffrable qu'à la condition qu'il existe une règle permettant de distinguer de façon univoque les mots les uns des autres.

Il est possible par exemple de transcrire une source symbolique Σ dans B^* selon un procédé régulier et déchiffrable. Pour cela il faut une règle qui permette de distinguer les encodages de deux symboles successifs. On en a donné un exemple dans la démonstration du théorème 1.1. Une autre possibilité consisterait à se donner un symbole spécial, par exemple $\chi = \#$, tel que $\# \notin \Xi$, et à construire la bibliothèque $\Xi' = \{\#\} \cup \Xi$. On peut imaginer la procédure suivante : Ξ étant énumérable, on affecte à chaque symbole de Ξ une chaîne unaire finie (par exemple la transcription en code unaire de son numéro d'indexation). On réserve au symbole $\#$ (transcrit par un 0) le rôle de séparateur entre chaque élément de Ξ (mis par exemple à la fin de sa description unaire).

A contrario, transcrire les symboles par l'équivalent en numération binaire de leur numéro d'indexation dans la bibliothèque est un procédé régulier, mais non déchiffrable (on ne peut pas distinguer deux nombres binaires successifs).

Il y a donc deux types de transcription, déchiffrable et non déchiffrable. Cela ne signifie pas qu'une transcription "non déchiffrable" ne soit pas utilisable : simplement, la séparation entre les mots, si elle existe, est effectuée par un moyen qui n'existe pas dans le code.

Cette distinction amène à considérer l'existence de plusieurs types de sources alphabétiques.

2.4.3. Source lexicale

**Définition 2.31*

Une *source lexicale*, que nous noterons C , est une source alphabétique dans laquelle la transcription des symboles en mots est effectuée selon un encodage non déchiffrable.

Cette source est munie de l'ordre numérique et de l'ordre lexicographique (d'où son nom).

Exemples : écriture binaire des entiers naturels ($1010011 = 10100.11 = 10.10011 = 10.100.11 = \text{etc...}$) ; une langue parlée, comme la langue française (enchanterions = en.chanterions = enchante.rions = en.chante.rions etc).

Une source lexicale ne peut donc pas produire de textes déchiffrables (lisibles en tant que suite de mots séparés) à l'aide de critères appartenant uniquement au système de codage, et ne peut pas produire de suites de suites de caractères. Inversement, un texte ne peut être lu que globalement, comme un mot nouveau. Donc une source lexicale ne produit que des mots.

Cela est toujours vrai d'une source à bibliothèque infinie : toute suite de caractères est un mot qui est la transcription d'un symbole de la bibliothèque. En revanche, si la bibliothèque de la source est finie, certaines suites de caractères, trop longues, ne sont pas des mots du langage et ne correspondent à aucun symbole. Donc :

* Propriété 2.1

La bibliothèque d'une source lexicale est infinie.

Dans ce cas, nous dirons que cette source est une source lexicale *bien formée*. A contrario, une source lexicale dont la bibliothèque serait finie sera dite "mal formée". Elle correspond à un code non déchiffrable inutilisable en tant que source de textes. En outre, elle suppose l'existence d'une information implicite permettant de limiter la longueur et donc le nombre des mots communicables (pour une discussion détaillée de ce point, voir §5.2.2).

Exemple

Considérons par exemple la bibliothèque infinie dénombrable des nombres premiers. Un procédé pourrait consister, en écriture, à transcrire un message constitué d'une suite de nombre premiers sous la forme du produit de ces nombres écrit en binaire. Le mot écrit est bien unique, et code bien un texte. Un tel code est non déchiffrable au sens où le nombre produit des nombres qui codent les mots ne permet pas de séparer ces mots. Mais il est quand même partiellement utilisable, car toute l'information n'est pas perdue : à la lecture, il suffit (cela est théoriquement possible...) de décomposer le mot reçu en son produit de facteurs premiers. Finalement, on a transmis par ce moyen l'ensemble des mots qui constituaient le texte initial, mais sans respecter leur ordre dans le texte : cette information est définitivement perdue.

On peut même diminuer encore les exigences que l'on porte sur l'encodage (binaire) en symbolisant l'ensemble objet par la bibliothèque formée des nombres premiers dont l'écriture binaire est symétrique (0,0), (1,1), (3,11), (5,101), (7,111), (17,10001), (31,11111). La source émettrice transcrit un texte sous la forme du nombre binaire produit des nombres de cette bibliothèque qui codent les mots du texte. Comme précédemment, la source réceptrice décompose le nombre reçu en facteurs premiers, l'information sur l'ordre des mots du texte est perdue, mais la liste des mots est conservée. En outre, l'encodage des mots étant symétrique, on se dispense d'une convention préalable implicite supplémentaire entre émetteur et récepteur concernant le sens de lecture des mots.

Cet exemple permet de préciser la notion de "code déchiffrable" : on demande à un tel

code de conserver la *structure* du texte, c'est-à-dire non seulement la liste des mots qui le composent (et donc le nombre d'occurrences de chacun d'eux), mais encore leur ordre dans le texte, donc de conserver à la fois la cardinalité (le nombre des mots) et l'ordinalité (leur ordonnancement), bref de conserver *l'énumération*.

Inversement, une source lexicale ne conserve que la cardinalité, information de dénombrement. Il faut donc regarder l'arbre de codage d'un tel code sous l'angle du comptage par dénombrement, et non de l'étiquetage par énumération. Tout nœud de l'arbre de codage est étiqueté par un mot-code. C'est le cas en particulier d'un père et d'un fils. Un message constitué du mot-code représenté par le fils est ambigu, puisqu'on peut l'interpréter comme ce mot lui-même, ou comme la concaténation du mot-code représenté par le père suivi d'un caractère (i.e. l'étiquette d'un fils de la racine).

Sous cet angle, l'arbre de codage est inutilisable en tant que règle de lecture. Il n'y a pas d'algorithme opérationnel de décodage des messages. Tous les mots-code sont équivalents, a priori équiprobables. La structure du langage est en fait une *étoile*, c'est-à-dire un arbre de profondeur unité, où aucun mot (i.e. aucun nombre entier naturel) ne contient "plus d'information" qu'un autre.

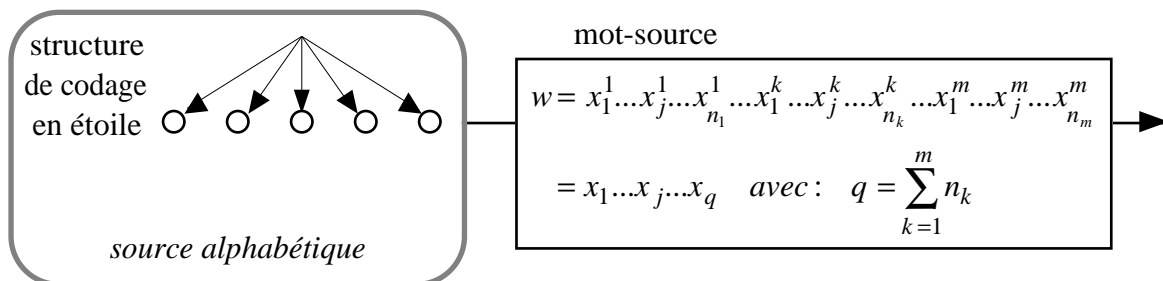


Figure 2.3 : la structure d'une source lexicale est "en étoile".

2.4.4. Source faible

*Définition 2.32a

Une source *faible* est une source alphabétique dans laquelle la transcription des symboles en mots est effectuée selon un encodage singulier.

De même que la déchiffrabilité est une propriété de discernabilité des mots, la régularité est une propriété de discernabilité des symboles. Sous l'angle pratique, la situation devient inconfortable : il n'y a pas assez de mots pour transcrire les symboles ! La cardinalité de l'ensemble des mots (le langage $L \subset A^*$) est inférieure à N , cardinalité de l'ensemble des

symboles (la bibliothèque Ξ). Ce qui conduit à une autre définition possible des sources faibles :

**Définition 2.32b*

Une source *faible* est une source alphabétique telle que :

$$\text{card}(\mathbf{L}) < \text{card}(\Xi)$$

Propriété 2.2

Une source faible est telle que : $Q^{n-1} \leq N-1$.

□ L'homonymie entre certains mots implique que le nombre de mots est au plus égal à $N-1$ (rappelons que $N = \text{card}(\Xi)$). La longueur n des mots est donc au plus égale à $\lceil \log_Q(N-1) \rceil$.

▣

Corollairement, il n'est pas possible de concaténer les mots d'une source faible, car l'emploi de la concaténation permettrait de construire de nouveaux mots, donc un vocabulaire infini, ce qui contredirait par définition le concept de "source faible" et permettrait de résoudre le problème de l'homonymie. La notion de source faible implique donc qu'on ne considère que des messages de longueur unité ($m = 1$) : la question de la déchiffabilité ne se pose pas.

De façon assez surprenante, bien que "faible", une telle source peut encore transmettre de l'information.

Exemples

- Soit une bibliothèque finie de N nombres entiers $n = \{1, 2, \dots, N\}$. On calcule pour chaque nombre $n \leq N$ la liste de ses diviseurs premiers. On code n par le produit de ceux-ci . Par exemple les nombres 15, 45, 75, 225, etc (tous les multiples par 3 ou par 5 de 15) sont codés par le mot-code 15 (3x5). Il s'agit donc bien d'une source faible (code singulier) : une infinité d'entiers ont la même transcription. A la détection, un calculateur pourrait décomposer le code reçu en facteurs premiers, puis calculer la liste des nombres-sources possibles, c'est-à-dire la liste exhaustive de tous les homonymes du mot-code reçu, jusqu'à N .

La "liste des ingrédients" intervenant dans la fabrication d'un produit alimentaire (sans précision des pourcentages ni de l'ordre dans lequel ils sont utilisés) contient ce genre d'information. On peut modéliser cette liste en supposant que l'on transcrit les quantités (en

pourcentage ou en valeur absolue) par la répétition des mots, chaque mot codant un type d'ingrédient sous la forme d'un nombre premier (écrit en binaire par ex.). L'ordre d'apparition des mots est l'ordre d'utilisation des ingrédients dans la recette. Si l'on transcrit cette suite par une multiplication, la divisibilité du résultat par chaque facteur premier conduit à la liste des ingrédients présents dans la recette... malgré l'homonymie (car cela revient à coder les messages "une pincée de sel" et "deux pincées de sel", c'est-à-dire "une pincée de sel plus une pincée de sel", par le même code. Le mot-code n'informe pas sur la quantité, mais sur la présence de sel). Une transcription opérée à partir de la propriété de divisibilité d'un nombre par des nombres premiers constitue ainsi un encodage singulier, porteur d'une information faible mais non nulle.

- Il y a bien d'autres façons de construire des sources faibles. Travailler par exemple en arithmétique modulo M , avec $M < N$, est une autre façon de "limiter" le langage à une taille inférieure à celle de la bibliothèque. On peut aussi ignorer le sens d'écriture (de gauche à droite ou de droite à gauche) des chaînes de caractères qui représentent les nombres : deux chaînes symétriques représentent le même nombre. Comme nous l'avons vu, le sous-ensemble des nombres premiers symétriques en écriture binaire constitue une source lexicale. Mais globalement, l'ensemble de ces nombres binaires, eu égard à l'ambiguïté du sens de lecture, constitue une source faible.

En résumé, les sources faibles constituent des systèmes de traitement de l'information dont les règles de calcul sont plus faibles que celles de l'arithmétique ordinaire parce qu'on code plusieurs nombres à l'aide d'une même chaîne de caractères (arithmétique modulo M , produits de nombres premiers, écriture non orientée des nombres, etc). Ces sources sont en quelque sorte "non-gödéliennes", car elles reposent sur des systèmes plus faibles que l'arithmétique.

2.4.5. Source préfixée

Nous allons maintenant porter notre attention sur les textes réellement réguliers et déchiffrables. Là encore, on rencontre immédiatement plusieurs solutions pour répondre à un tel cahier des charges. Par exemple on peut utiliser un caractère spécial pour marquer la fin des mots. On peut aussi se limiter à des mots de longueur constante. La séparation des mots successifs d'un texte est alors immédiate. Mais cela restreint le domaine de codage à un sous-ensemble fini A^n de A^* . Ce type de codage (bien que majoritairement employé) est limité aux langages finis. En outre il ne répond pas au critère de compressibilité maximale qui fonde la

notion de contenu d'information, car la longueur des mots-code est constante.

Au contraire, il existe des codes, dits "préfixés" ou "autodélimités", qui répondent à cette exigence.

définition 2.33

Un code *préfixe* est un code où aucun mot-code n'est le début d'un autre mot-code (règle dite "du préfixe").

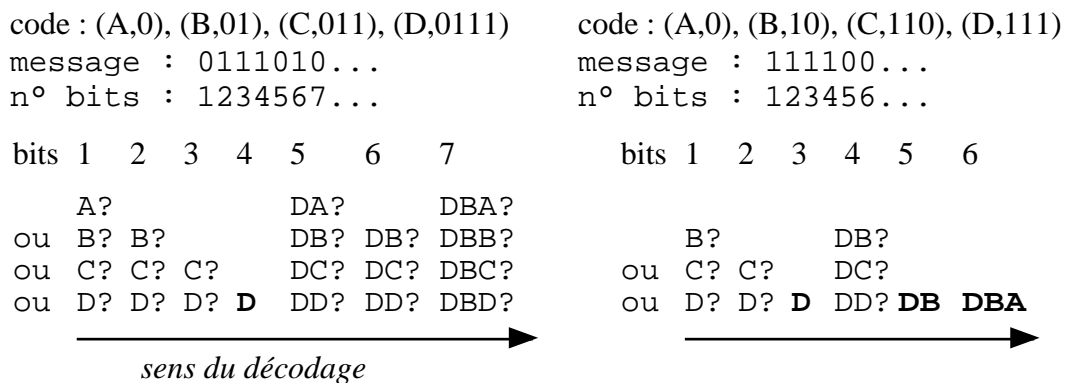
Propriété 2.2

Un code préfixe est déchiffrable.

- Soient un texte constitué de deux mots v et w , dont les codes sont respectivement x et y . Aucun mot n'étant le début d'un autre, on a nécessairement : $\mathcal{E}(x.y) = E(v).E(w)$, car le mot $x.y$ n'est pas un mot du code. Par récurrence on étend ce raisonnement à un texte t , tel que $\mathcal{E}(t.y) = E(x_1)...E(x_n).E(w)$



Dans l'arbre de codage ne sont utilisés pour représenter les mots-code que les nœuds extérieurs. Il n'est pas nécessaire de connaître la fin du message pour en décoder le début. Un code préfixe est décodable "à la volée", de façon instantanée (cf fig. 2.4b). C'est pourquoi un code préfixe est encore appelé code *instantané* ou code *irréductible*.



• Figure 2.4a : code non préfixé 2.4b code préfixé

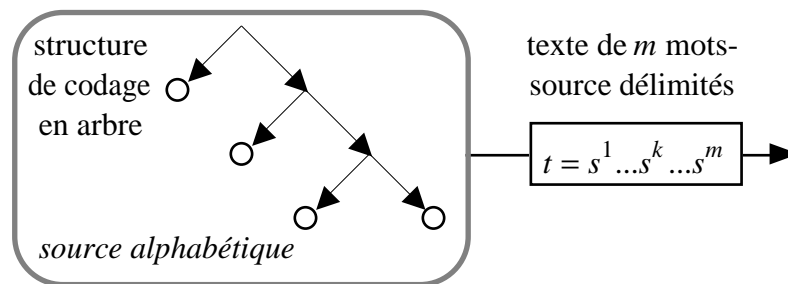
Parmi toutes les sources alphabétiques déchiffrables possibles, nous nous limiterons aux sources employant un code préfixe, dont les propriétés auront des conséquences fondamentales dans la suite de cette étude.

***Définition 2.34**

Une *source préfixée* K est une source dont les symboles sont transcrits alphabétiquement selon un encodage déchiffrable préfixé.

Cette source est munie de l'ordre numérique, de l'ordre lexicographique et de l'ordre préfixe (qui est un ordre partiel).

Exemple : écriture unaire des entiers naturels (suites de 1), avec séparateur terminal (0).



• Figure 2.5 : source préfixée

Cas particulier

Un cas particulier de source préfixée est celui où la valence de l'alphabet est supérieure ou égale à la cardinalité de la bibliothèque : $Q \geq N$. Dans le cas de l'égalité (resp. inégalité), il existe une bijection (resp. injection) qui revient à confondre la bibliothèque avec l'alphabet (resp. un sous-ensemble de l'alphabet) en un seul et unique ensemble. On construit ainsi une source symbolique.

Réciproquement, la longueur n des mots est égale à 1, aussi aucun "mot" ne peut être le préfixe d'un autre : une source symbolique est nécessairement de type préfixé. Cette propriété des sources symboliques s'étend au cas infini (dénombrable).

Conséquence : on a vu plus haut qu'une source lexicale bien formée est nécessairement à bibliothèque infinie (propriété 2.1). On vérifie ici une propriété complémentaire :

***Propriété 2.3**

L'alphabet d'une source lexicale bien formée est borné.

□ Si bibliothèque et alphabet étaient infinis (dénombrables), on pourrait toujours affecter à chaque symbole de la bibliothèque un caractère de l'alphabet, ce qui ferait de la source une source préfixée.

▣

2.4.6. Transcodage d'une source alphabétique vers une autre

Revenons à la notion d'information définie à partir d'un processus de compression : étant donné une source transcrite dans un certain langage, on cherche à coder les mots de ce langage (mots-source) en un autre langage (mots-code) tel que les mots-code soient de longueur inférieure aux mots-source. Deux sortes de sources font quatre types distincts de codage (i.e. transcodage). On note par une flèche (\rightarrow) le sens du codage : mot-source vers mot-code.

Type $C_C : C \rightarrow C$

Une source lexicale est codée en une source lexicale.

La description d'un nombre est un nombre. Dans ce cas, on ne se préoccupe pas de la délimitation ni des mots-source, ni des mots-code. Le but n'est donc pas d'échanger un texte, mais de partager un mot (car, faute de délimitation, on ne peut transmettre un texte).

Type $C_K : K \rightarrow C$

Une source préfixée est codée en une source lexicale.

La description d'un mot est un nombre. Un mot-source délimité est codé en un mot-code non délimité : par exemple numéroter en binaire naturel les mots d'un texte déchiffrable. Ici aussi l'échange de texte par le biais d'une suite d'entiers naturels est impossible, faute de délimitation.

Type $K_C : C \rightarrow K$

Une source lexicale est codée en une source préfixée.

La description d'un nombre est un mot. Un mot-source non délimité est codé en un mot-code délimité : par exemple écrire le programme (autodélimité) qui calcule le nombre source. Mais il n'est pas possible d'écrire un programme qui calculerait une suite de nombres-source concaténés, faute de délimiteur. L'échange des textes est toujours impossible.

Type $K_K : K \rightarrow K$

Une source préfixée est codée en une source préfixée.

La description d'un mot est un mot. Un mot-source délimité est codé en un mot-code délimité. C'est le cas (seul possible) d'une traduction ou d'un changement de vocabulaire, avec ou sans changement d'alphabet, dans le but d'échanger un texte.

Tous ces procédés de codage supposent l'existence d'un *système de réécriture*, mais seul le dernier cas permet de conserver la structure d'un texte.

2.4.7 Multicrucialité des textes

Cette constatation amène à penser "qu'un texte est plus qu'une suite de mots", car le premier contient une information implicite que ne contient pas la seconde. Une suite (quelconque) de mots est un *vecteur à une dimension*, c'est-à-dire une succession linéaire de chaînes pourvu d'un et un seul ordre, qui est l'ordre d'énumération des mots, dont découle l'énumération des caractères. Un texte bien formé implique la discernabilité des mots. Un texte est donc un *tableau* (vecteur à deux dimensions) car :

-on peut énumérer le texte dans l'ordre naturel de la succession des mots suivant une dimension :

$$x_1^1 \dots x_j^1 \dots x_{n_1}^1 \quad x_1^2 \dots x_j^2 \dots x_{n_2}^2 \quad \dots \quad x_1^k \dots x_j^k \dots x_{n_k}^k \quad \dots \quad x_1^m \dots x_j^m \dots x_{n_m}^m$$

-on peut aussi l'énumérer selon un tableau à deux dimensions :

$$\begin{array}{c} x_1^1 \dots x_j^1 \dots x_{n_1}^1 \\ x_1^2 \dots x_j^2 \dots x_{n_2}^2 \\ \dots \\ x_1^k \dots x_j^k \dots x_{n_k}^k \\ \dots \\ x_1^m \dots x_j^m \dots x_{n_m}^m \end{array}$$

Dans ce tableau, soit n_{max} la longueur du mot le plus long. L'alphabet étant binaire et les mots préfixés, chaque mot commence par exemple par un 1 et est autodélimité. On peut donc construire un tableau de n_{max} colonnes et m lignes en remplissant les cases vides par des 0. Connaisant n_{max} et m (que l'on peut coder selon un processus autodélimité), on peut énumérer le tableau dans le sens (conventionnel de gauche à droite et de haut en bas) lignes/colonnes selon une suite à une dimension de $m \cdot n_{max}$ caractères séparable en m blocs de n_{max} caractères, ou dans le sens colonnes/lignes selon une suite de n_{max} blocs de m caractères.

Exemple : cas d'un texte codé en unaire, dont chaque mot est terminé par un 0. Ainsi le texte 317124 codé 11101011111101011011110 ($n_{max} = 8$, $m = 6$) peut s'écrire sous la forme (les zéros mis en complément sont en gras) :

```
11100000
10000000
11111110
10000000
11000000
```

11110000

00000000

soit : 11100000100000001111111010000000110000001111000000000000

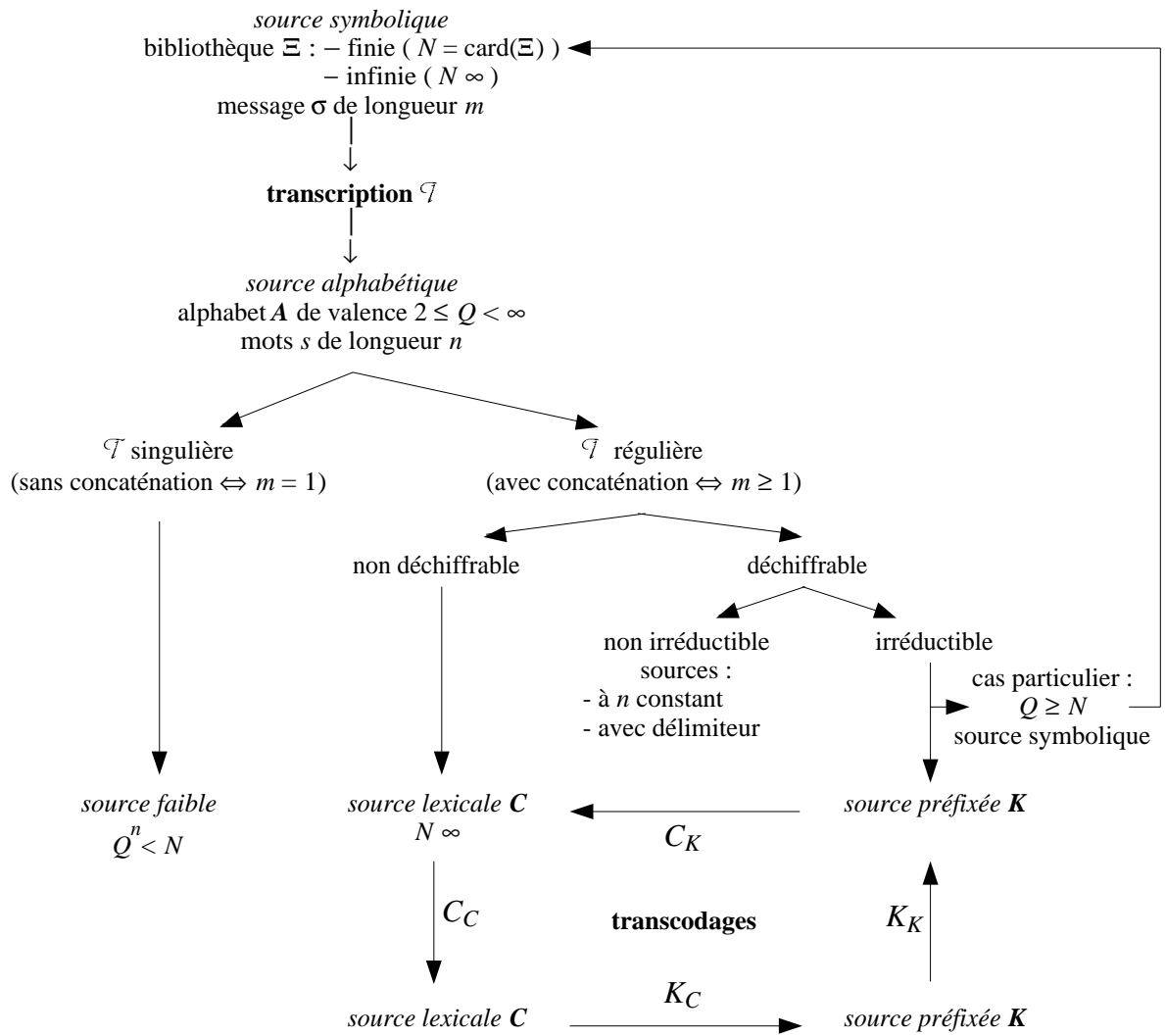
ou : 1111101010110101001000100100010000001000000100000000000

A cause de la redondance introduite (mots de longueur constante) on ne se situe plus ici dans le cadre de la compression d'information : la présentation d'un texte en tableau n'évalue pas le contenu d'information d'un objet. Cette présentation permet en revanche de compenser des erreurs de transmission de façon très efficace. C'est le cas par exemple des "turbo codes" [Berrou *et al.*, 1998] ou, tout simplement, des mots croisés. Nous avons cité plus haut un autre exemple, archétypique, qui est celui de la table des éléments chimiques. Nous avons appelé *multicrucialité* cette propriété des ensembles informationnels doublement ordonnés qui leur confère non seulement des propriétés de détection et de correction d'erreurs, mais encore de prédiction.

On peut donc s'attendre à ce qu'un transcodage de type K_K présente des propriétés mathématiques différentes des autres types de transcodage, et certainement plus puissantes, car les contraintes et les possibilités y sont plus grandes.

2.4.8. Résumé : classification des sources

L'ensemble des définitions proposées dans cette section est résumé figure 2.6.



• Figure 2.6 : classification des sources d'après les relations de transcription et de transcodage.

2.5. Extensions au continu de la théorie de l'information

Ce qui précède repose essentiellement sur la notion de dénombrabilité. La question se pose de savoir ce qu'il advient de ces définitions lorsqu'on part d'objets non dénombrables.

(i) Les objets

Par rapport à la définition 2.1, nous avons maintenant :

*Définition 2.36

Au sens de l'extension au continu de la théorie de l'information, un *objet* \mathcal{O} est un ensemble d'éléments notés o .

L'extension au continu modifie considérablement les caractéristiques de la théorie :

- \mathcal{O} reste un objet composite, c'est-à-dire constitué d'un ensemble d'éléments. On conserve la notion de cardinalité : soit $\Lambda = \text{card}(\mathcal{O})$. L'ordre, s'il existe, pourrait par exemple être un ordre similaire à celui de \mathbb{R} .

- les éléments o formant \mathcal{O} ne sont plus séparables, discernables les uns des autres (par définition du continu).

(ii) Les symboles

On peut toutefois conserver l'hypothèse d'une bijection entre \mathcal{O} et Ξ , ce dernier ensemble étant lui aussi supposé avoir la cardinalité du continu.

(iii) Les mots

- On conserve par hypothèse la notion d'alphabet A comme ensemble dénombrable fini de valence Q . Mais maintenant on suppose que la bibliothèque de symboles est un ensemble continu : $\Xi = \{\chi\}$. Un signe est une valeur prise dans cet ensemble. Etant donné un alphabet A , la transcription d'un signe en un mot produit une suite infinie de caractères (exemple : écriture binaire des nombres réels). Un message σ est une suite finie dénombrable de signes, comme précédemment, et est donc analogue à une fonction échantillonnée. Une telle source alphabétique sera dite "échantillonnée".

*Définition 2.35

Une source *échantillonnée* est une source alphabétique telle que : la bibliothèque n'est pas dénombrable, un message est une suite finie dénombrable de signes, la valence de l'alphabet

est $Q \in \mathbb{N}$, la transcription alphabétique de chaque signe est un mot formé d'une suite infinie dénombrable de caractères.

Nous verrons que la notion d'échantillonnage implique la notion de quantification et donc de résolution. Sur une suite infinie de caractères notée ω , transcription d'un réel $\in [0,1]$, la suite $\omega_{1:n}$ des n premiers caractères est l'information extraite de ω avec la précision Q^{-n} .

- Dans une source échantillonnée, un mot est une suite, finie ou infinie dénombrable, de caractères constituant un sous-ensemble de A^* . L'ensemble des mots n'est pas dénombrable. Mais une hypothèse paraissant raisonnable consiste à supposer que : bibliothèque et langage ont même cardinalité (continue).

Inversement, on peut imaginer qu'un mot soit une suite finie de caractères. Alors le langage est dénombrable, et l'on a :

$$\text{card}(\mathbf{L}) < \text{card}(\Xi)$$

Ce qui définit une source faible (def. 2.32b). Nous trouverons cette forme de source à propos de la corrélation quantique.

2.6. Conclusion : sources conjointes

Les différents types de sources qui viennent d'être décrits l'ont été en ne considérant *qu'une source à la fois*. Or nous avons souligné en introduction combien il paraissait important de considérer l'information, et donc sa source, non pas "dans l'absolu", indépendamment du contexte, de l'environnement, mais au contraire en fonction de, relativement à ce qui entoure une telle source d'information. Le modèle minimum des processus informationnels est donc un *couple* de sources conjointes, et un cas particulièrement simple est de considérer, a priori, que les deux membres du couple jouent des rôles équivalents. Nous reviendrons sur les conditions exactes qui permettent d'affirmer ou non cette équivalence entre les deux sources. Quoiqu'il en soit, ce schéma suppose d'imaginer l'existence, comme nous l'avons fait en introduction, d'au moins deux mécanismes élémentaires à la racine de tout processus informationnel : d'une part la *disjonction*, qui permet ou non d'effectuer une distinction entre les symboles ou une distinction entre les mots ; d'autre part la *conjonction*, mécanisme qui permet de concaténer des symboles pour former des messages, ou plus simplement d'associer des sources par la mise en commun des ressources sous-jacentes à un même message.