LE FORMALISME INFORMATIQUE

"Les théories doivent être aussi simples que possible, mais pas simplistes."

Einstein

Tout ce qui suit n'est qu'un rappel de mathématiques très simples car pour cette sensibilisation à l'IA il n'est pas nécessaire de manipuler des abstractions très élaborées.

Les ensembles

On entend par *ensemble* toute collection ou tout assemblage d'objets appelés *éléments* d'ensemble.

L'ensemble des trois premiers entiers naturels se notera {0,1,2}.

L'ensemble ne contenant aucun élément est noté Φ.

La notation d'appartenance $a \in E$ indique que a est un élément de l'ensemble E, ou que a appartient à E.

On dit que S est une *partie* ou encore un *sous-ensemble* d'un ensemble E et on note $S \subset E$ si et seulement si tout élément de S est un élément de E:

$$S \subset E \Leftrightarrow [\forall a,\ a \in S \Rightarrow a \in E]$$

Un sous-ensemble S de E se définit souvent par une propriété quelconque que seuls les éléments de S possèdent, par exemple le sous-ensemble des entiers naturels pairs:

$$\{a \in \mathbf{N} \mid a \text{ est pair}\}$$

Deux ensembles sont égaux ou coïncident si et seulement si tout élément de chacun d'eux appartient à l'autre:

$$E = F \Leftrightarrow E \subset F \text{ et } F \subset E$$

Un certain nombre d'opérations peuvent être définies sur les ensembles, regardons les plus courantes:

On appelle *intersection de deux ensembles* l'ensemble:

$$E \cap F = \{a \mid a \in E \text{ et } a \in F\}$$

On appelle *réunion de deux ensembles* l'ensemble:

$$E \cup F = \{a \mid a \in E \text{ ou } a \in F\}$$

On appelle *différence de deux ensembles* l'ensemble:

$$E - F = \{a \mid a \in E \text{ et } a \notin F\}$$

On appelle *produit cartésien de deux ensembles* l'ensemble de couples:

$$E \times F = \{(a,b) \mid a \in E \text{ et } b \in F\}$$

Une *application* d'un ensemble E dans un ensemble F est un "moyen" de faire correspondre à tout élément de E un unique élément de F.
L'ensemble des couples (a,b) où a décrit E et b est le correspondant de a par l'application est un sous-ensemble du produit cartésien E x F.

Les relations binaires

Définitions

Une *relation binaire* **R** sur deux ensembles A et B est un sous-ensemble du produit cartésien A x B.

$$\mathbf{R} \subset A \times B$$

Dans le cas A = B la relation binaire est dite dans A:

$$\mathbf{R} \subset A \times A = A^2$$

Une relation binaire **R** dans A est dite *réflexive* si tout élément de A est en relation avec lui-même:

$$\forall a \in A,\ (a,a) \in \mathbf{R}$$

La relation "être aussi grand que" dans un ensemble de personne est réflexive.

Une relation binaire **R** dans A est dite *symétrique* si pour tout élément a de A en relation avec un élément b de A alors b est aussi en relation avec a:

$$\forall (a,b) \in \mathbf{R},\ (b,a) \in \mathbf{R}$$

La relation "être le frère de" dans un ensemble de personnes est symétrique.

Une relation binaire **R** dans A est dite *antisymétrique* si pour tout élément a de A en relation avec un élément b de A avec b ≠ a, alors b n'est pas en relation avec a:

$$\forall(a,b),\ (a,b)\in\mathbf{R} \text{ et } a \neq b \ \Rightarrow\ (b,a) \notin \mathbf{R}$$

La relation "être le fils de" dans un ensemble de personne est antisymétrique.

Une relation binaire **R** dans A est dite *transitive* si pour tout élément a de A en relation avec un élément b de A et si b est en relation avec un élément c de A, alors a est en relation avec c:

$$\forall(a,b,c),\ (a,b)\in\mathbf{R} \text{ et } (b,c)\in\mathbf{R} \ \Rightarrow\ (a,c)\in\mathbf{R}$$

Si la proposition "les amis de mes amis sont mes amis" est vraie alors la relation "être amis de" dans un ensemble de personnes est transitive.

Une relation binaire **R** dans A est dite *d'équivalence* si et seulement si elle est réflexive, symétrique et transitive.

$$\mathbf{R} \text{ est d'équivalence} \Leftrightarrow \mathbf{R} \text{ est} \begin{matrix}\text{Réflexive}\\ \text{Symétrique}\\ \text{Transitive}\end{matrix}$$

La relation d'égalité "=" dans l'ensemble des nombres naturels **N** est d'équivalence.

Une relation binaire **R** dans A est dite *d'ordre partiel* si et seulement si elle est réflexive, antisymétrique et transitive.

$$\mathbf{R} \text{ est d'ordre partiel} \Leftrightarrow \mathbf{R} \text{ est } \begin{matrix}\text{Réflexive}\\ \text{Antisymétrique}\\ \text{Transitive}\end{matrix}$$

La relation "être un ascendant de" dans un ensemble de personnes est d'ordre partiel.

Une relation binaire **R** dans A est dite *d'ordre total* si et seulement si elle est d'ordre partiel et quels que soient deux éléments a et b de A il existe au moins une relation entre eux.

$$\mathbf{R} \text{ est d'ordre total} \Leftrightarrow \mathbf{R} \text{ est et } \begin{matrix}\text{d'ordre partiel}\\ \forall a,b\in A\ (a,b)\in\mathbf{R}\\ \text{ou}\\ (b,a)\in\mathbf{R}\end{matrix}$$

La relation "être inférieur ou égal à (≤)" dans **N** est d'ordre total.

S'il est possible de définir une relation d'ordre dans un ensemble A, deux relations d'ordre classiques peuvent être déduites dans A^2:

l'*ordre des coordonnées*, défini par:

$$(a,b) \leq (c,d) \Leftrightarrow a \leq c \text{ et } b \leq d$$

l'*ordre lexicographique*, défini par:

$$(a,b) \leq (c,d) \Leftrightarrow (a < c) \text{ ou } (a = c \text{ et } b \leq d)$$

Une *chaîne* dans une relation binaire **R** est une séquence de a_1 vers a_n $(a_1,a_2,...,a_n)$ avec $n \geq 1$ et pour tout $i \in \{1,...,n-1\}$ $(a_i,a_{i+1}) \in \mathbf{R}$.

Les définitions et propriétés des relations n-aire ne sont pas présentées ici, dans la mesure où il est relativement facile de les deviner en partant de celles des relations binaires.

Propriété de fermeture

La propriété de fermeture d'un sous-ensemble sous une relation est une notion qu'il est utile de bien maîtriser car elle est fondamentale dans la construction des langages, comme nous allons le voir plus loin.

Un sous ensemble $B \subset A$ est dit être *fermé* sous la relation $\mathbf{R} \subset A^{n+1}$, avec $n \geq 0$, **R** étant une (n+1)-aire relation dans A si:

$$\forall c \in B \text{ et } (b_1,b_2,...,b_n) \in B^n,\ (b_1,b_2,...,b_n,c) \in \mathbf{R}$$

La *fermeture réflexive et transitive* d'une relation binaire **R** dans A est notée $\mathbf{R}^*$ et on peut en déduire:

$$\mathbf{R}^* = \mathbf{R} \cup \{(a,b) \in A^2 \mid \exists \text{ une chaîne dans } \mathbf{R} \text{ de a vers b}\}$$

L'ensemble des entiers naturels **N** est sa propre fermeture sous la relation associée à la loi d'addition des entiers.

L'ensemble des entiers relatifs **Z** est la fermeture de **N** sous la relation associée à la loi de soustraction des entiers.

Structures à une loi

Une application * de A^2 dans A s'appelle une *loi de composition*. L'ensemble A est dit muni d'une loi. La structure ainsi formée est notée (A,*).

Une autre façon de voir une loi de composition est de considérer la loi comme étant une partie du cube cartésien A^3:

$$*((a,b)) = c \quad \text{notée} \quad a * b = c$$

$$\text{le composé c est unique avec } (a,b,c) \in A^3$$

La loi de composition étant une *application*, il faut que tout couple ait un composé.

Une loi est dite *commutative* si pour tout couple (a,b) de A^2

$$a * b = b * a$$

Une loi est dite *associative* si pour tout triplet (a,b,c) de A^3

$$(a * b) * c = a * (b * c)$$

Une loi est dite unitaire d'*élément neutre* ε si pour tout élément a de A

$$a * \varepsilon = \varepsilon * a = a$$

Un élément a de A est dit *inversible* s'il existe un élément b de A, noté a^{-1}, tel que

$$a * b = b * a = \varepsilon$$

Un *semi-groupe* est un ensemble muni d'une loi associative.

Un *monoïde* est un semi-groupe unitaire.

Un *groupe* est un monoïde où tout élément est inversible.

Le créateur de la théorie des groupes est le mathématicien français E. Galois (1811-1832), les autres grand noms associés à cette théorie sont H. Abel (1802-1829), A. Cauchy (1789-1857), C. Jordan (1838-1922), L. Sylow (1832-1918), F. Klein (1849-1925) et beaucoup de contemporains.

Un groupe commutatif est souvent appelé un groupe abélien.

Les langages

Les alphabets

Un alphabet Σ est un ensemble fini d'éléments appelés symboles.

{a,b,c,...,z} est l'alphabet latin.

{0,1} est l'alphabet binaire.

Un *mot* sur un alphabet est une séquence finie de symboles de l'alphabet.

L'ensemble de tous les mots, en incluant le mot vide noté ε, sur un alphabet Σ est noté Σ^*.

Σ^* est un ensemble infini dénombrable.

La longueur d'un mot m, notée |m|, est le nombre de symboles qui le composent.

Une fonction de position pos_m pour un mot m peut être définie:

$$\forall m \in \Sigma^* \quad \exists\ pos_m: \{1,2,...,|m|\} \Rightarrow \Sigma$$

Un élément $a \in \Sigma$ est une occurrence de $m \in \Sigma^*$ si

$$\exists\ j \in \{1,2,...,|m|\} \mid pos_m(j) = a$$

Le classement des mots dans un dictionnaire se fait grâce à l'ordre lexicographique.

Deux mots sur un même alphabet Σ peuvent être combinés pour en former un troisième par la *loi de concaténation* o. La structure ainsi formée (Σ^*, o) est un monoïde d'élément neutre le mot vide ε.

La concaténation de zéro mot est le mot vide ε, la concaténation de un mot est le mot lui-même.

Les langages formels

La définition d'un langage sur un alphabet est extrêmement simple, c'est certainement pour cela qu'elle est aussi puissante:

Tout sous-ensemble de l'ensemble des mots (Σ^*) formés sur un alphabet Σ sera appelé un *langage*.

Σ^*, Σ, Φ (l'ensemble vide) sont des langages sur Σ.

{soupe,aime,toto} est un langage sur l'alphabet latin.

Un langage étant un ensemble, nous pouvons appliquer toutes les opérations sur les ensembles pour créer d'autres langages.

Des opérations spécifiques peuvent être définies:

Le langage L sera dit *concaténation des langages* L1 et L2 si tout mot du langage L peut être vu comme la concaténation d'un mot du langage L1 et d'un mot du langage L2:

$$L = L1 \circ L2 = \{m \mid m = m1 \circ m2 \text{ avec } m1 \in L1 \text{ et } m2 \in L2\}$$

Une autre opération spécifique aux langages est la *fermeture* ou *Kleene star* d'un langage L, notée L^*. L^* est l'ensemble de tous les mots obtenus par concaténation de zéro ou plus de mots de L.

$$L^* = \{m \in \Sigma^* \mid m = m1 \circ m2 \circ \ldots \circ mn \text{ avec } n \geq 0 \text{ et } m1, m2, \ldots, mn \in L\}$$

Langages réguliers

L'ensemble de tous les langages possibles sur un alphabet donné Σ est 2 à la puissance Σ^* (l'ensemble des parties de Σ^*), qui est un ensemble infini indénombrable.

Nous ne pouvons donc pas étudier tous les langages possibles sur un alphabet donné; nous nous intéresserons à ceux qui nous semblent les plus simples, et dans un premier temps à ceux qui présentent une construction régulière.
Pour ce faire nous allons bâtir un sur-ensemble de l'alphabet Σ sur lequel nous construirons un langage de description d'expressions régulières. En étant un peu plus formel, cela donne:

Une expression régulière sur un alphabet Σ est un mot sur l'alphabet $\Sigma \cup \{[,], \cup, *, \Phi\}$:

$$\text{expression régulière} \in (\Sigma \cup \{[,], \cup, *, \Phi\})^*$$

Essayons de construire, ce nous appellerons le langage du contrôleur de billets L, qui est, il faut le souligner, une caricature du langage de contrôleur, défini comme suit:

Devant chaque personne le contrôleur se fait un devoir de commencer la conversation (fortement limitée à cet ensemble de mots sur l'alphabet:
Σ = {billet?, S'il_vous_plaît, merci, bonjour,
ça_va_vous_coûter_cher, au_revoir})
par un bonjour et la terminer par un au_revoir ou bien encore par aucune formule de politesse si le passager a été fortement désagréable.
Il est évident que l'ensemble des propos tenus par ce contrôleur va présenter une structure régulière tout au long de la journée.

$$L = \{bonjour\} \circ [\{billet?\}^* \cup \{S'il\_vous\_plait\}^* \cup \{ça\_va\_vous\_couter\_cher\}^* \cup \{merci\}^*] \circ [\{au\_revoir\} \cup \Phi^*]$$

Ce qui en simplifiant les notations donnera:

L = bonjour
[billet?* ∪ S'il_vous_plait^* ∪
ça_va_vous_couter_cher* ∪ merci*]
[au_revoir ∪ Φ*]

Prenons un langage plus informatique, mais tout aussi régulier L dont les mots sur l'alphabet binaire présentent trois ou quatre occurrences de 1:

L = {0}* o {1} o {0}* o {1} o {0}* o {1} o {0}* o [[{1} o {0}*] ∪ Φ*]

et donc, par simplification:

L = 0*10*10*10*[[10*]∪ Φ*]

Nous arrivons à une définition simple des langages réguliers:

Un langage est dit *régulier* s'il peut être décrit par une expression régulière.

Il existe plus d'une représentation régulière représentant le même langage régulier.

Les expressions régulières sont pratiques pour représenter certains langages infinis.

Attention: évidemment tous les langages ne sont pas réguliers.

Les langages indépendants du contexte

Vous êtes capable de *reconnaître* la phrase "L'intelligence artificielle nécessite plus de math que je pensais !", vous agissez donc comme un analyseur de langage; et si vous êtes capable de *produire* la phrase "je commence à en avoir ras-le-bol du formalisme", vous agirez, alors, comme un générateur de langage.

Les expressions régulières peuvent être vues comme des générateurs de langages, dans la mesure où les mots sont produits de gauche à droite.

Nous allons, maintenant nous intéresser à un certain type de générateurs de langages, loin d'être comparable à ceux du langage naturel, mais déjà plus évolués que les expressions régulières: les grammaires indépendantes du contexte (en anglais « context free »).

Auparavant, essayons de voir comment à partir d'une expression régulière nous pouvons produire un mot du langage; pour ce faire prenons un exemple:

{0 [0 ∪ 1]* 1}

Soit L le langage sur l'alphabet binaire dont les mots commencent par un 0 et se terminent par un 1.

$$L = \{0\} \circ [\{0\} \cup \{1\}]^* \circ \{1\}$$

Tout mot M du langage est de la forme 0A1 où A est un mot d'un autre langage $[\{0\} \cup \{1\}]^* = \Sigma^*$, pour traduire cette proposition nous écrirons:

$$M \Rightarrow 0A1$$

où $\Rightarrow$ veut dire "peut être de la forme". Cette expression est appelée une *règle de production* (c'est une règle de grammaire, comme nous le verrons plus loin).

Que vaut A? A est un mot sur Σ^*, donc peut être de la forme 0A ou 1A ou bien encore ε ce qui se traduit par:

$$A \Rightarrow 0A$$
$$A \Rightarrow 1A$$
$$A \Rightarrow \varepsilon$$

Nous avons donc fait correspondre au langage L quatre règles de productions, regardons leur utilisation dans l'exemple:

Production du mot 011011

$M \Rightarrow 0A1$	0A1
$A \Rightarrow 1A$	01A1
$A \Rightarrow 1A$	011A1
$A \Rightarrow 0A$	0110A1
$A \Rightarrow 1A$	01101A1
$A \Rightarrow \varepsilon$	011011

Une grammaire context free est un générateur de langage qui opère comme cela vient d'être fait dans l'exemple précédent.

Pourquoi ce type de générateur est-il appelé indépendant du contexte? A chaque étape de production (remplacement du membre gauche de la règle utilisée par le membre droit) nous n'avons pas besoin de savoir ce qui s'est passé avant l'application de cette règle ni de ce qui se passera après, donc l'application d'une règle est indépendante de son contexte d'utilisation.

Donnons maintenant, une définition rigoureuse d'une grammaire context-free, en considérant qu'une règle $A \Rightarrow B$ peut être considérée comme un couple (A,B):

Une *grammaire indépendante du contexte* G est un quadruplet (Σ, T, R, I)

où

Σ est un alphabet

$T \subset \Sigma$ est le sous-ensemble des terminaux

$R \subset \Pi \times \Sigma^*$ est un sous ensemble de règles, avec $\Pi = \Sigma - T$ l'ensemble des non-terminaux

$I \in \Pi$ symbole initial

On dira qu'un mot $m2 \in \Sigma^*$ *dérive* d'un mot $m1 \in \Sigma^*$ dans la grammaire G, la dérivation sera notée

$$m1 \overset{G}{\Rightarrow} m2$$

si et seulement si il existe des mots m, m', m" et un non-terminal A tels que

$$m1 = mAm',\ m2 = mm"m' \text{ et } A \Rightarrow m"$$

La relation $\overset{G}{\Rightarrow}$ dans $\Sigma^* \times \Sigma^*$ peut être étendue par fermeture.

Notons

$\overset{*G}{\Rightarrow}$ la fermeture réflexive et transitive de $\overset{G}{\Rightarrow}$.

Nous pouvons désormais écrire simplement le langage L généré par une grammaire indépendante du contexte (Σ,T,R,I):

$$L = \{m \in T^* \mid I \overset{*G}{\Rightarrow} m\}$$

qui veut simplement dire que les mots d'un langage généré par une grammaire indépendante du contexte, sont tous les mots qui dérivent du symbole initial par fermeture réflexive et transitive.

Une grammaire indépendante du contexte G est dite régulière si et seulement si le membre droit de chaque règle contient au plus un non-terminal, qui, s'il est présent doit être le dernier symbole du mot.

$$G = (\Sigma,T,R,I) \text{ régulière} \Leftrightarrow$$

$$R \subset (\Sigma - T) \times T^* \circ ((\Sigma - T) \cup \{\varepsilon\})$$

Nous avons maintenant une autre façon de voir les langages réguliers car:

Un langage est régulier si et seulement si il est généré par une grammaire régulière.

Les langages générés par un système de réécriture

Jusqu'à présent nous avons rencontré deux types de langages: les langages réguliers et les langages indépendants du contexte, et nous avons vu que les premiers étaient des cas particuliers des seconds. Définissons maintenant un générateur de langage plus général dépendant du contexte:

Une *grammaire* (aussi appelée système de réécriture ou grammaire non-restrictive) G est un quadruplet (Σ, T, R, I)

où

Σ est un alphabet

$T \subset \Sigma$ est le sous-ensemble des terminaux

$R \subset (\Sigma^* \circ \Pi \circ \Sigma^*) \times \Sigma^*$ est un sous-ensemble de règles,
avec $\Pi = \Sigma - T$ l'ensemble des non-terminaux

$I \in \Pi$ symbole initial

De la même façon que pour les grammaires context free, nous pouvons écrire simplement le langage L généré par un système de réécriture (Σ, T, R, I):

$$L = \{m \in T^* \mid I \overset{*G}{\Rightarrow} m\}$$

les mots d'un langage généré par un système de réécriture, sont tous les mots qui dérivent du symbole initial par fermeture réflexive et transitive.

Toute grammaire indépendante du contexte est un système de réécriture.

> Par exemple la règle $m_1 A m_2 \Rightarrow m_1 m_3 m_2$ peut être remplacée par la proposition "remplacer A par m_3 dans le contexte où m_1 est à gauche de A et m_2 est à droite de A".