Langages

I Alphabet, lettres, mots, langages

I.1 Définitions

Définition 1 : Alphabet et lettres

- Dans ce chapitre et le suivant, on appelle alphabet un ensemble fini non vide Σ.
- Les éléments de Σ sont appelés des lettres.

Exemple 1 : On prendra souvent dans la suite un alphabet à deux lettres et en particulier $\Sigma = \{a, b\}$.

À noter : les "lettres" peuvent être bien être des objets abstraits, ou par exemple des chiffres.
On peut tout à fait considérer l'alphabet $\Sigma = \{0, 1\}$.

Implémentation : pour les implémentations, on se limitera au cas où Σ est composé de symboles ascii et on l'implémentera par le type `char`.

Définition 2 : Mot sur Σ

- On appelle mot sur Σ une suite finie de lettres de Σ.
- Pour des raisons qui s'éclaireront dans la suite, on note Σ^* l'ensemble des mots sur Σ.
- Pour $u \in \Sigma^*$, on appelle longueur de u et on note $|u|$ le nombre de lettres du mot u.

Exemples et conventions 2 :

1. L'unique mot de longueur 0 est un mot sur Σ. On l'appelle le mot vide et on le note ε.
2. Un mot de longueur 1, c'est-à-dire formé d'une seule lettre, est un mot de Σ. Par abus, on identifie un tel mot avec son unique lettre. Pour $\Sigma = \{a, b\}$, on a donc exactement deux mots de une lettre dans Σ^* : a et b.
3. En général, pour $|u| = n$ on notera simplement $u = a_0 a_1 \cdots a_{n-1}$ où les a_i sont les lettres composant u.
4. Y a-t-il plus de mots de longueur 3 sur $\Sigma = \{a, b\}$ ou plus de mots de longueur 2 sur $\Sigma = \{a, b, c\}$?

Implémentation : on pourra implémenter les mots de principalement deux façons :

— par le type `string` ;

— par le type `char list`.

Définition 3 : Préfixe, suffixe, facteur

Soit $u = a_0 a_1 \cdots a_{n-1}$.

- Un mot de la forme $a_0 a_1 \cdots a_{j-1}$ avec $j \leqslant n$ est appelé un préfixe de u.
- Un mot de la forme $a_i a_{i+1} \cdots a_{n-1}$ avec $i \leqslant n$ est appelé un suffixe de u.
- Un mot de la forme $a_i a_{i+1} \cdots a_{j-1}$ avec $i \leqslant j \leqslant n$ est appelé un facteur de u.

Exemple 3 : On implémente ici les mots par le type `string`. On rappelle qu'en Caml, on peut comparer l'égalité de deux `string` avec `=`, c'est une opération de coût linéaire en la longueur des `string` comparées.

1. Écrire une fonction permettant de tester si un mot `f` est ou pas facteur d'un mot `u`.
2. Quelle est sa complexité en fonction de $|u|$ et $|m|$? *Ce n'est pas très efficace...*

Un des objectifs des *automates* (voir le chapitre suivant) est de déterminer efficacement si un mot, ou plus généralement un *motif* (voir la section II.2) apparaît comme facteur d'un texte, que l'on voit lui-même comme un très long mot.

Définition 4 : Langage

On appelle langage un ensemble de mots, c'est-à-dire une partie de Σ^*.

L'idée est de dire qu'une suite quelconque de lettres ne constitue pas nécessairement un mot bien formé. On a un ensemble de règles qui indiquent quels mots sont bien formés et quels mots ne le sont pas, et les mots bien formés sont ceux de notre langage.

Exemples 4 :

1. Langues naturelles (aspect lexical) : Σ est l'alphabet habituel, par exemple l'alphabet latin, et L est l'ensemble des mots de du dictionnaire de la langue naturelle considérée (français, anglais, italien,...). Un tel langage est très simple à décrire car **il est fini** : il peut donc être donné en extension.
2. Langues naturelles (aspect syntaxique) : Σ est l'ensemble des mots d'une langue naturelle (par exemple le français), et L est l'ensemble des phrases correctes que l'on peut former dans cette langue naturelle. Un tel langage est beaucoup plus complexe à décrire et n'est jamais fini. On a longtemps essayé d'identifier les règles communes propres à ce type de langage pour faire de la traduction automatique mais à ma connaissance ça n'a jamais abouti et finalement on utilise d'autres méthodes pour la traduction automatique.
3. Langage des préfixes d'un mot : étant donné un mot $u \in \Sigma^*$, l'ensemble des préfixes de u forme un langage fini de cardinal .....
4. Quelques langages triviaux à connaître sur Σ :
   - Le langage vide $\emptyset$: aucun mot.
   - Le langage $\{\varepsilon\}$: un seul mot, le mot vide.
   - Le langage Σ^* : tous les mots sont dans le langage.
   - Le langage Σ : les mots du langage sont les mots de longueur 1 (qu'on identifie aux lettres).

I.2 Monoïde $(\Sigma^*, \cdot)$

Définition 5 : Concaténation des mots

Soient $u = a_0a_1\cdots a_{n-1}$ et $v = b_0b_1\cdots b_{m-1}$ deux mots. On appelle concaténation de u et v le mot $a_0a_1\cdots a_{n-1}b_0b_1\cdots b_{m-1}$ qu'on peut noter $u\cdot v$ ou plus simplement uv.

Remarques 1 :

- L'opération $\cdot$ est associative.
- L'opération $\cdot$ a un élément neutre : .....
- Un mot p est un préfixe d'un mot u si et seulement si .....
- Un mot s est un suffixe d'un mot u si et seulement si .....
- Un mot f est un facteur d'un mot u si et seulement si .....

Définition 6 : Puissances d'un mot

Soient $u \in \Sigma^*$ et $n \in \mathbb{N}$ deux mots. On appelle puissance $n^{\text{ième}}$ de u et on note u^n le mot $\underbrace{u\cdot u\cdot\cdots\cdot u}_{n \text{ fois}}$.

Exemples 5 :

1. Quel que soit le mot u, on a $u^0 =$ .....
2. Quel que soit l'entier n, on a $\varepsilon^n =$ .....
3. Un exemple idiot : $(aba)^3 =$ .....

Exemple 6 : Un exercice classique mais pénible : deux mots u et v commutent si et seulement si ils sont puissances d'un même mot. Le sens réciproque est facile, ainsi que le sens direct si u ou v est vide. Pour le reste, on raisonne par récurrence forte sur $|u|+|v|$. Tu peux le détailler comme exercice bonus.

I.3 Opérations sur les langages

Les opérations qui nous intéresseront sur les langages sont les suivantes :

Définition 7 : Opérations régulières sur les langages

Soit L un langage, soient L_1, L_2 deux langages.

- La réunion de L_1 et L_2 est le langage $L_1 \cup L_2$.
- La concaténation de L_1 et L_2 est le langage $L_1\cdot L_2 = \{uv,\ u\in L_1,\ v\in L_2\}$.
- Pour $n\in\mathbb{N}$, la puissance $n^{\text{ième}}$ de L est le langage $L^n = \{u_1u_2\cdots u_n,\ u_i\in L\}$.
- L'étoile de Kleene de L est le langage $L^* = \bigcup_{n\in\mathbb{N}} L^n = \{u\in\Sigma^*,\ \exists n\in\mathbb{N},\ \exists u_1,\dots,u_n\in L,\ u = u_1u_2\cdots u_n\}$.

Exemples 7 :

1. On peut voir Σ^* comme l'étoile de Kleene du langage des mots de longueur 1 sur Σ, qu'on note bien Σ.
2. Pour tout langage L on a $L\cup L =$ ..... et $L\cdot\{\varepsilon\} =$ ..... = ...
3. Pour tout langage L on a $L^0 =$ ..... et $L^1 =$ .....
4. Pour $n\in\mathbb{N}$ on a $\emptyset^n = \begin{cases}$

5. Décrire simplement $\{a\}^* \cdot \{b\}^*$.

6. Décrire simplement $\{a\}^* \cdot \{b\} \cdot \{a\}^*$.

7. Décrire simplement $\{a\}^* \cdot \{a,b\}^* \cup \{a,b\}^* \cdot \{b\}^*$.

Remarque 2 : Pour désambiguër certaines expressions sans parenthèses, on considère l'opération $*$ sur les langages comme prioritaire sur l'opération $\cdot$, qu'on considère elle-même comme prioritaire sur l'opération $\cup$.

II Langage rationnel ou régulier

II.1 Langages rationnels

Définition 8 : Langages rationnels

On définit l'ensemble des langages rationnels sur Σ par induction comme suit.
L'ensemble $\mathcal{L}$ des langages rationnels sur Σ est le plus petit ensemble de langages tel que :

1. $\emptyset \in \mathcal{L}$ et $\{\varepsilon\} \in \mathcal{L}$;
2. pour tout $a \in \Sigma$, $\{a\} \in \mathcal{L}$;
3. $\mathcal{L}$ est stable par réunion $(\forall L_1, L_2 \in \mathcal{L},\ L_1 \cup L_2 \in \mathcal{L})$;
4. $\mathcal{L}$ est stable par concaténation $(\forall L_1, L_2 \in \mathcal{L},\ L_1 \cdot L_2 \in \mathcal{L})$;
5. $\mathcal{L}$ est stable par passage à l'étoile de Kleene $(\forall L \in \mathcal{L},\ L^* \in \mathcal{L})$.

Exemples 8 :

1. Tout singleton de Σ^* est un langage rationnel : pourquoi ?

2. Tout langage fini est rationnel : pourquoi ?

3. Σ^* est un langage rationnel : pourquoi ?

Exemple 9 : On a déjà vu que l'ensemble des préfixes d'un mot était un langage fini : c'est donc un langage rationnel. Plus généralement, l'ensemble des préfixes des mots d'un langage rationnel L est lui aussi un langage rationnel. C'est un résultat très simple en utilisant les automates, mais difficile à établir à l'aide de la seule définition.
La définition précédente est une définition "par le haut" : on n'a pas défini ce qu'était un langage rationnel. On a défini ce qu'est l'ensemble des langage rationnel comme le plus petit ensemble stable par certaines propriétés. Comme toujours dans ce type de situations [1] on peut aussi donner une définition "par le bas", qui explicite plus clairement comment construire des langages rationnels (normalement, on a commencé grâce aux exemples à entrevoir cette définition par le bas).

II.2 Langages réguliers

Définition 9 : Expression régulières

On définit l'ensemble des expressions régulières sur Σ par induction comme suit.
L'ensemble des expressions régulières est le plus petit ensemble tel que :

1. $\varnothing$ et ε sont des expressions régulières ;
2. pour tout $a \in \Sigma$, a est une expression régulière ;
3. étant données deux expressions régulières e_1 et e_2, l'expression $(e_1|e_2)$, qu'on peut aussi noter $(e_1 + e_2)$, est une expression régulière ;
4. étant données deux expressions régulières e_1 et e_2, l'expression $(e_1 \cdot e_2)$, qu'on peut aussi noter $(e_1 e_2)$, est une expression régulière ;
5. étant donnée une expression régulière e, l'expression (e^*) est une expression régulière.

L'objectif de cette définition est de donner un sens précis à la notion de *motif* évoquée plus haut. Lorsqu'on est amené à chercher dans un texte une expression donnée, cette expression n'est pas nécessairement un mot précis.

Par exemple, je veux savoir si un texte parle d'animaux : je peux chercher la chaîne de caractère "animal" dans le texte, mais je vais manquer les occurrences de "animaux" ; je peux chercher la chaîne de caractère "animaux" dans le texte, mais je vais manquer les occurrences de "animal" ; enfin je peux chercher la chaîne de caractère "anima" dans le texte, mais je risque d'obtenir trop de réponses qui ne m'intéressent pas comme "animation".
Ce que je veux chercher dans ce texte, c'est l'expression régulière $\boxed{\textbf{anima(l|ux)}}$**.**

1. Exemples analogues mais issus du cours de mathématiques : sous-espace vectoriel engendré par une partie ; enveloppe convexe d'une partie ; idéal engendré par un élément...

Remarque 3 : Pour désambiguër certaines expressions sans parenthèses, on considère l'opération $*$ sur les expressions régulières comme prioritaire sur l'opération $\cdot$, qu'on considère elle-même comme prioritaire sur l'opération $+$.

Au lieu de parler de mots dénotés par un motif, on parle de langage dénoté par une expression régulière, mais l'idée reste qu'une expression régulière peut désigner simultanément toutes sortes de mots, et peut donc représenter un langage.

Définition 10 : Langage régulier

- On définit le langage $L(e)$ dénoté par l'expression régulière e par induction sur e :
  1. $L(\varnothing) = \emptyset$; $L(\varepsilon) = \{\varepsilon\}$;
  2. pour tout $a \in \Sigma$, $L(a) = \{a\}$;
  3. pour toutes expressions régulières e_1 et e_2, $L(e_1 + e_2) = L(e_1) \cup L(e_2)$;
  4. pour toutes expressions régulières e_1 et e_2, $L(e_1 e_2) = L(e_1)L(e_2)$;
  5. pour toute expression régulière e, $L(e^*) = L(e)^*$.
- Par abus, on pourra noter plus simplement e (plutôt que $L(e)$) le langage dénoté par une expression régulière e.
- Un langage est dit régulier lorsqu'il est dénoté par une expression régulière.

Exemples 10 : On peut reprendre les trois derniers des exemples 7 avec le formalisme des expressions régulières : cela évite d'écrire des accolades :

Théorème 1

Un langage est rationnel si et seulement s'il est régulier.

C'est très général : la définition de langage régulier est la définition "par le bas" naturellement associée à la définition d'un langage rationnel (qui elle est une définition "par le haut"). Je m'apprête pour ainsi dire à copier-coller la démonstration de ce que l'ensemble des combinaisons linéaires qu'on peut former avec un ensemble de vecteurs coïncide avec le plus petit sous-espace vectoriel contenant cet ensemble.

Démonstration.

$\boxed{\Rightarrow}$ Montrons que l'ensemble des langages rationnels est inclus dans l'ensemble des langages réguliers.
Comme l'ensemble des langages rationnels est le plus petit qui vérifie les 5 propriétés de la définition 8, il suffit de voir que l'ensemble des langages réguliers vérifie ces 5 propriétés :

1. $\emptyset$ est bien régulier puisqu'on a $\emptyset = L(\varnothing)$, et $\{\varepsilon\}$ est bien régulier puisqu'on a $\{\varepsilon\} = L(\varepsilon)$.
2. pour tout $a \in \Sigma$, $\{a\}$ est bien régulier puisqu'on a $\{a\} = L(a)$;
3. soient $L(e)$, $L(e')$ deux langages réguliers, $L(e) \cup L(e')$ est bien régulier puisqu'on a $L(e) \cup L(e') = L(e + e')$;
4. soient $L(e)$, $L(e')$ deux langages réguliers, $L(e) \cdot L(e')$ est bien régulier puisqu'on a $L(e) \cdot L(e') = L(ee')$;
5. soit $L(e)$ un langage régulier, $L(e)^*$ est bien régulier puisqu'on a $L(e)^* = L(e^*)$.

$\boxed{\Leftarrow}$ Montrons que l'ensemble des langages réguliers est inclus dans l'ensemble des langages rationnels, c'est-à-dire que pour toute expression régulière e, $L(e)$ est rationnel. On procède par induction sur e :

1. Si $e = \varnothing$ alors on a bien $L(e) = \emptyset$ qui est rationnel, et si $e = \varepsilon$ alors on a bien $L(e) = \{\varepsilon\}$ qui est rationnel.
2. Si e est de la forme a avec $a \in \Sigma$, alors $L(e) = \{a\}$ est bien rationnel.
3. Si e est de la forme $e' + e''$ avec e' et e'' des expressions régulières telles que $L(e')$ et $L(e'')$ sont rationnels, alors $L(e) = L(e') \cup L(e'')$ est bien rationnel.
4. Si e est de la forme $e'e''$ avec e' et e'' des expressions régulières telles que $L(e')$ et $L(e'')$ sont rationnels, alors $L(e) = L(e') \cdot L(e'')$ est bien rationnel.
5. Si e est de la forme e'^* avec e' une expression régulière telle que $L(e')$ est rationnel, alors $L(e) = L(e')^*$ est bien rationnel.

Voilà. Bof.

Application 1 :

Pour $\Sigma = \{a, b\}$, donner sans démonstration des expressions régulières qui dénotent les langages suivants :

1. l'ensemble des mots qui contiennent au moins un a :
2. l'ensemble des mots qui contiennent au plus un a :
3. l'ensemble des mots dans lesquels que toute série de a est de longueur paire :
4. l'ensemble des mots dont la longueur n'est pas divisible par 3 (beurk, un peu) :

On en déduit que ces langages sont rationnels !

II.3 Langages locaux

Les langages locaux ont la propriété d'être essentiellement caractérisés par trois de leurs caractéristiques :

1. l'ensemble des lettres qui peuvent s'obtenir comme première lettre d'un mot du langage ;
2. l'ensemble des lettres qui peuvent s'obtenir comme dernière lettre d'un mot du langage ;
3. les facteurs de longueur 2 des mots du langage.

Étant donné un langage L sur un alphabet Σ, on définit :
• $P_1(L) = \{a \in \Sigma, a\Sigma^* \cap L \neq \emptyset\} = \{a \in \Sigma, \exists u \in \Sigma^*, a.u \in L\}$
• $S_1(L) = \{a \in \Sigma, \Sigma^* a \cap L \neq \emptyset\} = \{a \in \Sigma, \exists u \in \Sigma^*, u.a \in L\}$
• $F_2(L) = \left\{u \in \Sigma^2, \Sigma^* u \Sigma^* \cap L \neq \emptyset\right\} = \left\{u \in \Sigma^2, \exists v, w \in \Sigma^*, v.u.w \in L\right\}$
On a écrit P_1 pour "préfixes de longueur 1", S_1 pour "suffixes de longueur 1", F_2 pour "facteurs de longueur 2".

Proposition-Définition 11 : Langage local
Soit L un langage. Les propriétés suivates sont équivalentes :
i/ $L \setminus \{\varepsilon\} = (P_1(L)\Sigma^* \cap \Sigma^* S_1(L)) \setminus \left(\Sigma^* \left(\Sigma^2 \setminus F_2(L)\right) \Sigma^*\right)$;
ii/ il existe trois ensembles $P \subset \Sigma$, $S \subset \Sigma$ et $F \subset \Sigma^2$ tels que : $L \setminus \{\varepsilon\} = (P\Sigma^* \cap \Sigma^* S) \setminus \left(\Sigma^* \left(\Sigma^2 \setminus F\right) \Sigma^*\right)$.

Démonstration. Il est évident que ii/ implique i/ !
Réciproquement supposons $L \setminus \{\varepsilon\} = (P\Sigma^* \cap \Sigma^* S) \setminus \left(\Sigma^* \left(\Sigma^2 \setminus F\right) \Sigma^*\right)$.
• Soit $u \in L \setminus \{\varepsilon\}$, on a en particulier $u \in P\Sigma^* \cap \Sigma^* S$ et donc $u \in P\Sigma^*$, si bien que la première lettre de u est dans P. Ceci montre $P_1(L) \subset P$. De plus pour $a \in P$ on a $a \in P\Sigma^* \cap \Sigma^* S$, et $a \notin \Sigma^* \left(\Sigma^2 \setminus F\right) \Sigma^*$ car cet ensemble ne contient aucun mot de longueur 1. Donc $a \in L$. Ceci montre $P \subset P_1(L)$, et on a donc finalement $P = P_1(L)$.
• On montre de façon identique $S = S_1(L)$, et de façon (seulement) analogue $F = F_2(L)$.

Éclairer la terminologie "langage local" en traduisant de façon plus intelligible la définition et en proposant un algorithme permettant de déterminer si un langage est local.

On verra dans le prochain chapitre que tout langage local est rationnel (la réciproque est fausse). On pourrait le faire dès à présent "à la main", mais c'est beaucoup plus facile en utilisant les automates.