Théorie des Langages Formels
Chapitre 1

Florence Levé

Florence.Leve@u-picardie.fr

Année 2022-2023

Au sujet de l'unité d'enseignement

La théorie des langages formels est une des matières fondamentales de l'informatique.

- Objectifs de l'enseignement :
  - comprendre les concepts de base de la théorie des langages formels ;
  - comprendre son rôle et son intérêt en informatique
  - savoir manipuler et utiliser langages, automates, grammaires.
- De nombreuses notions et approches de l'informatique vont être évoquées, notamment à travers les exemples :
  - algorithmique, complexité, preuve de programme, expressions régulières, ...
- Pré-requis : connaissance minimale de la notion d'ensemble
- Modalités de contrôle des connaissances :
  - partiel-examen

Bibliographie

- *Introduction à la calculabilité*, P. Wolper, Dunod 2006 (3ème édition), chapitres 1 à 4.
- *Théorie des automates (méthodes et exercices corrigés)*, P. Séébold, Vuibert 1999.
- *Méthodes mathématiques pour l'informatique* (4ème édition), J. Vélu, chapitres 21 et 22, Dunod 2005.
- *Théorie des langages et des automates*, J.-M. Autebert, Masson 1994 (deuxième partie, p41–67).
- *Éléments de théorie des automates*, J. Sakarovitch, Vuibert 2003 (chapitre 1, p55–232).

- Nombreux sites webs : n'hésitez pas à faire vos propres recherches.
- Ce cours est basé sur celui dispensé par Gwénaël Richomme jusqu'en 2009.

Qu'est-ce que les langages formels ?

- Un langage formel = un ensemble de mots.
- Exemples
  - L'ensemble des mots définis dans un dictionnaire.
  - L'ensemble des phrases que l'on peut écrire en français.
    - Remarque : alphabet = lettres, espace et symboles de ponctuation, ...
  - L'ensemble des programmes en langage JAVA.

Quelques problématiques

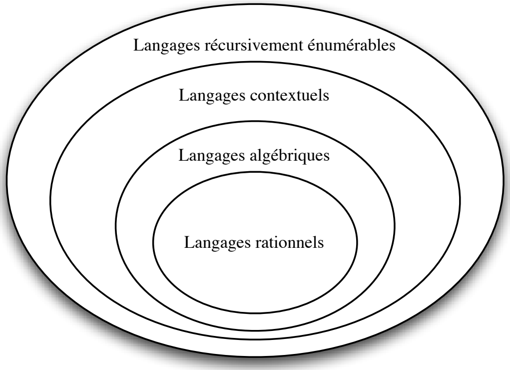
- Analyse lexicale : est-ce que mon programme utilise les mots de base du langage ?
  - utilisation d'automates dans les compilateurs.
- Analyse syntaxique : est-ce que les mots/phrases de mon programme sont correctement construits ?
  - utilisation de grammaires dans les compilateurs.
  - (problème : reconnaissance des langues naturelles)
- Est-ce que le programme fait ce que je veux ?
  - indécidable.
  - preuve à la main dans de nombreux cas !
- Quels langages peuvent être reconnus par une machine ?

Quelques origines

Des besoins similaires dans 3 thématiques :

- Informatique :
  - Besoin de décrire de maniére finie certains langages infinis
  - Premiers langages de programmation : algol, ...
  - Claude Shannon 1949 : description de protocoles de communication
- Logique :
  - Besoin de définir formellement le discours mathématique
  - Stephen Cole Kleene 1954 : montre qu'un langage est reconnaissable si et seulement s'il peut étre engendré, à partir des lettres de l'alphabet, à l'aide des trois opérations union, produit et étoile.
- Linguistique
  - Besoin de décrire les langues naturelles
  - Début des années 50 : premiéres tentatives visant à utiliser l'ordinateur pour traduire un texte
  - Noam Chomsky 1956 : hiérarchie des langages.

La Hiérarchie des langages de Chomsky



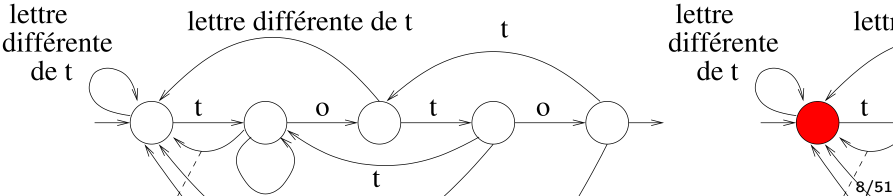
- {rationnels}
  - = {reconnaissables}
  - langages reconnus par un automate et/ou définis par une expression réguliére.
- $\subset$ {algébriques}

Quelques utilisations

1. Recherche de motif dans un texte

Texte en entrée : Une histoire de toto de plusUne histoire de toto

de plusUne histoire de toto de plusUne_ histoire de toto de plusUne histoire de toto de plusUne histoire de toto de plusUne histoire de toto de plusUne histoire de toto de plusUne histoire de toto de plusUne histoire de toto de plusUne histoire de toto de plusUne histoire de toto de plusUne histoire_ de toto de plusUne histoire de toto de plusUne histoire de toto de plusUne histoire de_ toto de plusUne histoire de toto de plusUne histoire de toto de plusUne histoire de toto de plusUne histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

- Applications de la recherche de mots/motifs
  - Recherche dans un index
    - index d'un fichier ;
    - index du web.
  - Recherche de virus
    - fichier : un mot (une suite) de 0 et de 1.
    - virus : un mot (ou un ensemble de mots)
    - base des signatures : un gros automate (version simplifiée)

Quelques utilisations

Exemple 2. Compilation

- Génération de compilateurs (et donc de nouveaux langages informatiques)
  - Automates : analyse lexicale ;
  - Grammaires : analyse syntaxique.
  - Remarque : la science de la compilation fait appel à des techniques supplémentaires : transformation de code, contréle de type, ...)

Quelques utilisations

3. REGEXP (REGular EXPressions)

- Expressions réguliéres
- Utilisées par les mécanismes de recherche/remplacement
  - éditeurs de texte : JEdit, emacs
  - commandes de base unix : ls, grep, sed, ...
  - inclus dans des langages de script : perl, javascript, ...

Quelques utilisations

4. Liens avec d'autres domaines

Les langages formels apparaissent ou sont liés à de nombreux domaines, de l'informatique ou non :

- réseaux, systémes d'exploitation, logiciels (compilation, traduction, vérification),
- modélisation, présentation de protocoles, algorithmes
- calculabilité, complexité, logique,
- combinatoire des mots, dynamique symbolique, théorie des nombres,
- linguistique (traitement de la langue naturelle),
- électronique,
- bioinformatique (séquenéage du génome),
- imagerie (analyse d'images),
- . . .

Le plan du cours

Un langage est un ensemble de mots. Pour étudier les langages, le plan du cours sera le suivant :

- Définitions, mots, langages, langages rationnels
- Automates et langages reconnaissables
  - définitions, fonctionnement d'un automate
  - équivalence avec langages rationnels
  - déterminisme
  - minimalité
- Langages non reconnaissable

Lettres et alphabets

Un langage est un ensemble de mots. Un mot est écrit avec des lettres appartenant à un alphabet.

- Lettre : symboles.
- Alphabet : ensemble fini non vide de lettres.

Exemples

1. Alphabet latin, grec, ...
2. Chiffres : $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \}$
3. Caractères ASCII, UNICODE, ...
4. Parties de ces alphabets : $\{a, b\}$, $\{0, 1\}$

Attention : les symboles doivent être non ambigüs !

$$\left.\begin{array}{l}\{a, b, ab\},\\ \{ab, a, ba\}\end{array}\right\} \text{ne sont pas des alphabets.}$$

Lettres grecques courantes

lettre	minuscule	majuscule
alpha	α	
beta	β	
gamma	γ	Γ
delta	δ	Δ
epsilon	ε	
phi	ϕ	Φ
psi	ψ	Ψ
rho	ρ	
mu	μ	
nu	ν	

Mots

- Mot : suite de lettres.
  - Notation : les lettres sont accolées.
  - Exemples : bonjour, abaababaab, 0110101, ...
  - Formellement, $a_1 \ldots a_n$ est le mot constitué dans l'*ordre* des lettres a_1 puis a_2 puis $\ldots a_n$.
  - On ne tient pas compte de la signification éventuelle.
- Mot vide : suite vide de lettres : ε.

Attention ! Ne pas confondre le mot vide ε avec l'ensemble vide $\emptyset$

La concaténation

- Définition : la concaténation de deux mots u et v est le mot obtenu en mettant bout à bout dans l'ordre les lettres de u puis les lettres de v.
- Notation : uv ou $u.v$
- La concaténation est aussi appelée produit de concaténation
- Exemple : $(aabac).(dab)$ vaut $aabacdab$
- Formellement, si $\begin{cases} n \geq 0 \text{ et } p \geq 0 \text{ sont deux entiers,} \\ a_1, \ldots, a_n,\ b_1, \ldots, b_p \text{ sont des lettres,} \end{cases}$

  alors

$$(a_1 \ldots a_n).(b_1 \ldots b_p) = a_1 \ldots a_n b_1 \ldots b_p$$

Monoïde

- Propriétés de la concaténation :
  - $u\varepsilon = u = \varepsilon u$
  - $(uv)w = u(vw)$
- Propriété : l'ensemble des mots muni de la concaténation forme un monoïde.
- Monoïde : ensemble muni d'une opération interne associative possédant un élément neutre.
- Autres exemples de monoïdes : $(\mathbb{N}, +)$, $(\mathbb{N}^+, \times)$
- L'ensemble des mots définis sur un alphabet A se note A^*

Monoïde libre

- Propriété fondamentale :
  Tout mot se décompose de manière unique sur les lettres.

  $a_1 \ldots a_n = b_1 \ldots b_p$ implique
  $$\begin{cases} n = p \\ \text{et } a_i = b_i \text{ pour tout } 1 \leq i \leq n. \end{cases}$$
- Le monoïde A^* est dit libre (de base A).

Longueur

- Longueur d'un mot : nombre de lettres qui le composent.
- Notation : $|u|$ est la longueur de u.
- Exemple : $|abaab| = 5$
- Propriétés :
  - $|\varepsilon| = 0$
  - $|uv| = |u| + |v|$

Nombre d'occurrences

- $|u|_a$: nombre d'occurrences de la lettre a dans un mot u
- Exemple : $|abaab|_a = 3$
- Propriétés :
  - $|\varepsilon|_a = 0$
  - $|uv|_a = |u|_a + |v|_a$
  - $|u| = \sum_{a \in A} |u|_a$

Un peu de vocabulaire

- Facteur : $u = pvs$
- Préfixe (facteur gauche) : dans l'exemple précédent, p, pv, sont des préfixes de u.
- Suffixe (facteur droit) : dans l'exemple précédent, s, vs sont des suffixes de u.
- Exemple : Facteurs, préfixes et suffixes du mot abaab ?

Ensembles

- Un ensemble est caractérisé par la notion d'appartenance
  - pour un ensemble E, tout objet x appartient ou non à E.
  - $x \in E$: x appartient à E
    (on dit aussi que x est dans E) ;
  - $x \notin E$: x n'appartient pas à E.

Définitions d'ensemble

- Par extension : en précisant les valeurs de l'ensemble entre accolades (valeurs séparées par des virgules).
  - Exemple : $E = \{a, c, f\}$
- Par compréhension, en précisant la propriété que vérifient les élements de l'ensemble.
  - Ensemble des entiers pairs =

$$\{x \in \mathbb{N} \mid x \bmod 2 = 0\}$$

  - Ensemble des mots sur A de longueur paire =

$$\{x \in A^* \mid |x| \bmod 2 = 0\}$$

Égalité

- Deux ensembles sont égaux si les éléments de l'un appartiennent à l'autre et réciproquement.
- En d'autres termes $E = F$
  - si pour tout x dans E, x appartient à F et pour tout x dans F, x appartient à E
  - (avec les notations de la logique) $(\forall x \in E, x \in F)$ et $(\forall x \in F, x \in E)$
- Exemples, avec a, b et c trois lettres différentes :
  - $\{ab, ac, a, b\} = \{a, ab, b, ac\}$
  - $\{a, bc\} = \{a, a, bc, a, bc, a\}$

  Attention
  - $\{ab, ac, a, b\} \neq \{ab, ac\}$
  - $\{ab, ac, b\} \neq \{ab, ac, a\}$

Notation de la logique

- Très utile pour synthétiser des idées/présentations !
- Un langage à apprendre !
- $\exists$ il existe
- $\forall$ pour tout
- $\Rightarrow$ implique : dire qu'une propriété p implique une propriété q *signifie* que si la propriété est vérifiée alors la propriété q l'est aussi
- (p $\Rightarrow$ q) est aussi une propriété (vraie si p est fausse)
- $\Leftrightarrow$ est équivalent à : dire qu'une propriété p équivaut à une propriété q *signifie* que p et q sont simultanément vraie ou simultanément fausse.

Inclusion

- Un ensemble E est dit inclus dans un ensemble F si tout élément de E appartient à F ($\forall x \in E, x \in F$).
  - Notation : $E \subseteq F$
- Un ensemble E est dit strictement inclus dans un ensemble F si $E \subseteq F$ et $E \neq F$.
  - Notation : $E \subset F$
- Exemples :
  - $\{ab, ac, a, b\} \subseteq \{a, ab, b, ac\}$ (en fait $=$)
  - $\{ab, ac\} \subset \{ab, ac, a, b\}$
  - $\{ab, ac, a, b\} \not\subset \{ab, ac\}$
- Propriété importante :

$$\boxed{X \subseteq Y \text{ et } Y \subseteq X \Leftrightarrow X = Y}$$

Ensemble vide

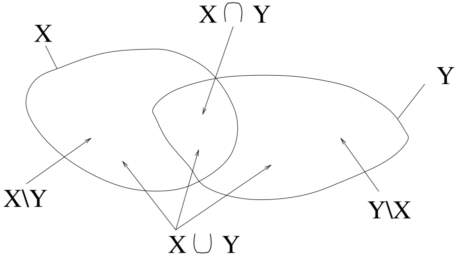
- L'ensemble constitué d'aucun élément.
- Notation : $\emptyset$.
- Exercice
  - ε :
    - le mot vide
    - d'ici à la fin de ce cours, l'expression rationnelle désignant l'ensemble $\{\varepsilon\}$
  - $\emptyset$: l'ensemble vide
  - $\{\varepsilon\}$: l'ensemble ayant comme seul élément le mot vide.
  - $\{\emptyset\}$: l'ensemble ayant comme seul élément l'ensemble vide.

Implémentation d'ensemble

- Tableaux pour ensembles finis, voire tableaux triés
  Mais il peut y avoir plus efficace : arbres, ...
- Automates
- ...

Opérations ensemblistes

- **Union.** $X \cup Y = \{x \mid x \in X \text{ ou } x \in Y\}$.
- **Intersection.** $X \cap Y = \{x \mid x \in X \text{ et } x \in Y\}$.
- **Complémentation.**
  $X \setminus Y = \{x \mid x \in X \text{ et } x \notin Y\} \quad (= X \setminus (X \cap Y))$.
- Diagramme de Venn :



Propriétés (1/4)

- $X \cup \emptyset = X$,
- $X \cap \emptyset = \emptyset$,
- $X \setminus \emptyset = X$,
- $\emptyset \setminus X = \emptyset$.
- $X \cup X = X$,
- $X \cap X = X$,
- $X \setminus X = \emptyset$.

Propriétés (2/4)

- (commutativité) $\begin{cases} X \cup Y = Y \cup X, \\ X \cap Y = Y \cap X. \end{cases}$
- Attention !
  $X \setminus Y$ n'est pas nécessairement égal à $Y \setminus X$
- Exemple $X = \{a\}$, $Y = \{b\}$.
  - $X \setminus Y = X = \{a\}$,
  - $Y \setminus X = Y = \{b\}$.

Propriétés (3/4)

- (associativité) $\left\{ \begin{array}{l} X \cup (Y \cup Z) = (X \cup Y) \cup Z, \\ X \cap (Y \cap Z) = (X \cap Y) \cap Z. \end{array} \right.$
- On peut donc enlever les parenthèses
- Attention ! on peut avoir $X \setminus (Y \setminus Z) \neq (X \setminus Y) \setminus Z$.

  Exemple : $X = \{a, b\}$, $Y = \{a\}$, $Z = \{a\}$:
  - $X \setminus (Y \setminus Z) = X$,
  - $(X \setminus Y) \setminus Z = \{b\}$

Propriétés (4/4)

- (distributivité) $\left\{ \begin{array}{l} X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z), \\ X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z). \end{array} \right.$

- Lois de de Morgan :
$\left\{ \begin{array}{l} X \setminus (Y \cap Z) = (X \setminus Y) \cup (X \setminus Z), \\ X \setminus (Y \cup Z) = (X \setminus Y) \cap (X \setminus Z). \end{array} \right.$

Produit cartésien

- Le produit cartésien sera utile pour définir les transitions possibles d'un automate.

- Soient $E_1, \ldots, E_n$ des ensembles, le produit cartésien de ces ensembles est égal à l'ensemble des n-uplets $(x_1, \ldots, x_n)$ tels que $x_i \in E_i$ pour chaque i, $1 \leq i \leq n$:

$$E_1 \times E_2 \times \ldots \times E_n = \{(x_1, \ldots, x_n) \mid x_i \in E_i, 1 \leq i \leq n\}$$

- Exemple : $\{a, b\} \times \{ab, ba, c\} =$

$$\{(a, ab), (a, ba), (a, c), (b, ab), (b, ba), (b, c)\}$$

- Exemple : $\{1, 2\} \times \{a, b\} \times \{2, 3\} =$

$$\{(1, a, 2), (1, a, 3), (1, b, 2), (1, b, 3), (2, a, 2), (2, a, 3), (2, b, 2), (2, b, 3)\}$$

Cardinal

- cardinal d'un ensemble fini = son nombre d'éléments.
- Notation : $Card(E)$ ou $\#E$
- Exemple :
  - $Card(\{a, b\}) = 2$
  - $Card(\{ab, ba, c\}) = 3$
  - $Card(\{a, b\} \times \{ab, ba, c\}) = 6$.
  - $Card(\{1, 2\} \times \{a, b\} \times \{2, 3\}) = 8$
- Pour $E_1, E_2, \ldots, E_n$ (n ≥ 1) des ensembles finis,

$$Card(E_1 \times E_2 \times \ldots \times E_n) = Card(E_1) \times Card(E_2) \times \ldots \times Card(E_n)$$

Opérations rationnelles

- Ce sont les opérations d'union, de produit et d'itération.

Produit

- Soient X, Y deux ensembles de mots, le produit (on dit aussi la concaténation) de X et Y est l'ensemble :

$$XY = \{xy \mid x \in X, y \in Y\}.$$

- Notation : XY ou $X.Y$,
- Exemple : $\{ab, aba, b\}\{ab, b\} = \{abab, abb, abaab, bab, bb\}$.
- Comme la concaténation de mots, le produit d'ensembles n'est pas commutatif.

  Exemple :
  - $baba \in \{ab, b\}\{ab, aba, b\}$.
  - $baba \notin \{ab, aba, b\}\{ab, b\}$.
  - $\{ab, b\}\{ab, aba, b\} \neq \{ab, aba, b\}\{ab, b\}$.

Produit – autres exemples

- Soit P l'ensemble des mots de longueur paire ($\varepsilon \in P$) :

$$PP = P$$

- Soit I l'ensemble des mots de longueur impaire.

$$I.I = P \setminus \{\varepsilon\}$$

$$IP = PI = I$$

Propriétés du produit

- si $X \subseteq T$ et $Y \subseteq Z$ alors $XY \subseteq TZ$
- $X(Y \cup Z) = XY \cup XZ$
- $(X \cup Y)Z = XZ \cup YZ$
- $X(Y \cap Z) \subseteq XY \cap XZ$
  En général pas égalité.
  Par exemple, $X = \{ab, a\}$, $Y = \{a\}$, $Z = \{ba\}$:
  - $X(Y \cap Z) = \emptyset$,
  - mais $XY \cap XZ = \{aba\}$.
- $(X \cap Y)Z \subseteq XZ \cap YZ$
- $\emptyset X = X\emptyset = \emptyset$
- $X\{\varepsilon\} = \{\varepsilon\}X = X$

Puissances

- Idée : $X^k = \overbrace{X...X}^{k \text{ fois}}$
- Inductivement $\begin{cases} X^0 = \{\varepsilon\}, \text{ et} \\ X^k = X^{k-1}X = XX^{k-1} \text{ quand } k \geq 1 \end{cases}$
- Définition par compréhension :
$$X^k = \{x_1 \dots x_k \mid x_i \in X, 1 \leq i \leq k\}$$
- (Convention usuelle : $x_1 \dots x_0 = \varepsilon$.)
- Exemples :
  - Soit A un alphabet, A^k est l'ensemble des mots de longueur k sur A.
    $\{a, b\}^3 = \{aaa, aab, aba, baa, abb, bab, bba, bbb\}$.
  - $\{a, ba, ab\}^2 = \{aa, aba, aab, baa, baba, baab, abba, abab\}$
  - Soit X l'ensemble des mots contenant au moins un a. Pour $k \geq 1$, X^k est l'ensemble des mots contenant au moins k occurrences de a.

Itération (Étoile)

- $X^* = \bigcup_{i=0}^{\infty} X^i$

$$X^* = \{x_1 \dots x_n \mid n \geq 0, x_i \in X\}$$

$$X^* = \{x_1 \dots x_n \mid n \geq 1, x_i \in X\} \cup \{\varepsilon\}$$

- Pour X l'ensemble des mots contenant au moins un a, $X^* = X \cup \{\varepsilon\}$.
- $\{aa, ab, ba, bb\}^*$ est l'ensemble des mots de longueur paire sur $\{a, b\}$.

Itération restreinte

- $X^+ = \bigcup_{i=1}^{\infty} X^i$

$$X^+ = \{x_1 \ldots x_n \mid n \geq 1, x_i \in X\}$$

Propriétés de l'itération

- pour i entier, $X^i \subseteq X^*$
- $\varepsilon \in X^*$, $X \subseteq X^*$
- $\emptyset^* = \{\varepsilon\}$
- si $X \subseteq Y$ alors $X^* \subseteq Y^*$
- $(X \cup \{\varepsilon\})^* = X^*$
- $(X^*)^* = X^*$
- $X^*X^* = X^*$
- $(X \cup Y)^* = ((X \cup \{\varepsilon\})(Y \cup \{\varepsilon\}))^* = (X^*Y^*)^*$
- $X^* = X^+ \cup \{\varepsilon\}$
- Si $\varepsilon \in X$, alors $X^* = X^+$. Dans ce cas, $X^+ \neq X^* \setminus \{\varepsilon\}$.
- $X^+ = XX^* = X^*X$

Langages rationnels/réguliers

- La famille des *langages rationnels* sur A est la plus petite (au sens de l'inclusion) famille de langages qui contient
  - l'ensemble vide,
  - les ensembles $\{a\}$ pour a lettre de A,

  et telle que, pour tous ensembles rationnels X et Y,
  - $X \cup Y$ est rationnel,
  - XY est rationnel et
  - X^* est rationnel.
- Notation : $\mathrm{Rat}(A^*)$ ou $\mathrm{Rat}(A)$,
- En d'autres termes, un ensemble est rationnel si et seulement s'il peut être obtenu à partir des ensembles finis en utilisant un nombre fini de fois les opérations d'union, de produit et d'itération.

Langages rationnels/réguliers – exemples

- ensemble vide
- les singletons $\{u\}$ pour $u \in A^*$ (u mot sur A)
- ensembles finis de mots
- A^*
- $\{aa, ab, ba, bb\}^* = (\{a, b\}\{a, b\})^* = (AA)^*$ avec $A = \{a, b\}$.
- $A^*\{a\}A^*$.

Expressions régulières

- Expressions formées par les règles suivantes :
  - $\emptyset$, ε et les éléments de A sont des expressions régulières.
  - Si E et F sont des expressions régulières, $(E+F)$, (EF) et $(E)^*$ sont des expressions régulières.
- En pratique, les parenthèses sont omises s'il n'y a pas d'ambigüité.
- La priorité des opérateurs est : "$*$", ".", "$+$".
- Des noms d'ensembles (représentables par une expression rationnelle) peuvent aussi être introduits dans les expressions rationnelles.
- Bien entendu, il ne faut pas que A contienne les symboles $(,), \emptyset, \varepsilon, *, +$, et le symbole de la virgule.
- Les notations présentées ci-après sont celles de la théorie des langages. Elles sont souvent différentes dans les logiciels utilisant les "regular expressions" (REGEXP).

Langage associé à une exp. rég

- Définition
  - $L(\emptyset) = \emptyset$;
  - $L(\varepsilon) = \{\varepsilon\}$;
  - $L(a) = \{a\}$ pour a lettre ;
  - $L(E + F) = L(E) \cup L(F)$;
  - $L(EF) = L(E)L(F)$;
  - $L(E^*) = L(E)^*$.
- Exemples
  - A^*aA^* désigne l'ensemble $A^*\{a\}A^*$
  - L'ensemble$\{aa, ab, ba, bb\}^*$ des mots de longueur paire :

$$(aa + ab + ba + bb)^*$$

Souvent ! (Notation pas très pure)

$$(aa, ab, ba, bb)^*$$

Le lien entre les derniers épisodes

- Lemme. Un langage est rationnel si et seulement si une expression régulière le représente.
- D'où la terminologie *régulière*.
- D'où, confusion usuelle entre le langage et son expression.

Compléments

- Attention, l'$*$ porte sur ce qui précède immédiatement !
  Par exemple, $aa^* \neq (aa)^*$, $ab^* \neq (ab)^*$
- Remarque. Une expression rationnelle ne contient pas, par déf., de notation de puissance et d'itération restreinte.
  Par souci de simplification, elles pourront toutefois être utilisées (sauf ambiguïté)
  Exemple d'utilisation de puissance :
  A^3 pour les mots de longueur 3 (au lieu de AAA)
- Attention ! $a^+a \neq a + a = \{a\} \cup \{a\}$.
- un même ensemble peut être représentée par plusieurs expressions rationnelles différentes.
  Exemple : $AA^* = A^*A$

Des questions. Introduction à la suite.

- Est-ce que tous les ensembles sont rationnels ?
  NON.
- Comment tester l'égalité de deux langages rationnels ?
  AUTOMATES
- Est-ce que l'intersection de deux langages rationnels est encore un langage rationnel ? Même question pour la complémentation.
  OUI
- Comment savoir si un langage est rationnel ?
  DEPEND DE LA REPRESENTATION DU LANGAGE
- Comment tester (en machine) si un mot appartient à un langage rationnel ?
  AUTOMATES