UNIVERSITÉ FRANÇOIS - RABELAIS TOURS

Automates

Théorie des langages

Damien Nouvel

Théorie des langages
Plan

- Alphabets et langages
- Expressions régulières formelles

Plan

- Alphabets et langages
- Expressions régulières formelles

- Structure du langage :
  - Algèbre, **ensemble** muni d'**opérations** (anneau)
  - « Briques de base » : **alphabet**
  - **Propriétés formelles** particulières du langage
- Différents manières de définir un langage :
  - Intentionnelle : « tous les mots qui… »
  - Extensionnelle : {mot, autremot, a, b, 3, 42 }
  - Définitoire : {xyz | x = yz }
  - **Opération sur d'autres langages** (union, concaténation, intersection, complémentaire …)

- **Alphabet**, un ensemble « hors-langage » :
  - Ensemble $\sum$ des symboles utilisés par un langage
    - $\sum$ = {a, b, c, d … z} (= [a-z])
    - $\sum$ = {0, 1, 2… 9} (= [0-9])
    - $\sum$ = {a, f, d, x}
    - $\sum$ = {b, c, 0, $, µ, z}
    - $\sum$ = {ab, de, xyz}
    - $\sum$ = {abc, a, tuv, vu}
  - On considère les éléments comme **atomiques** : « abc » peut-être un symbole unique pour un langage
  - Pas de répétitions au sein d'un alphabet (ensemble)

- **Mot** (ou **chaîne**) :
  - Un élément de l'alphabet est un mot
  - **Concaténation « . » de symboles** issus d'un alphabet :
    - $\sum = \{0, 1\}$: $\alpha_1 = 0.0.1.0$, $\alpha_2 = 1.0.0.0.1.0$ ...
    - Associative : (a.b).c = a.(b.c)
    - Non commutative : $0.1 \neq 1.0$
  - Quelque soit l'alphabet, il est possible de former le **mot** (ou chaîne) **vide**, noté « **ε** » (différent de ∅)
  - **Taille du mot** : | x |, nombre de **symboles** :
    - | 0.1.0.0 | = 4 ($\sum = \{0, 1\}$)
    - | d.d.ab.ab.d | = 5 ($\sum = \{ab, d\}$)
    - | ε | = 0 ($\forall \sum$)

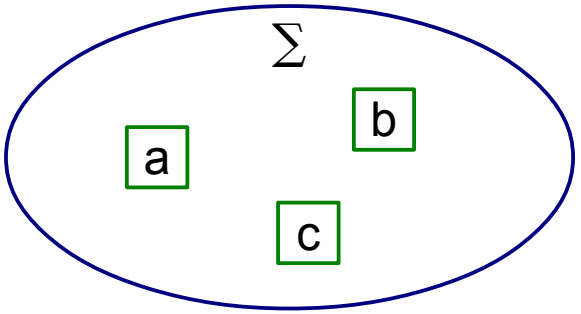
- Un langage est un **ensemble de mots**
- Définition de langages à partir d'**alphabets** :
  - Langage généré par un alphabet (récursive) :
    - Tout mot de l'alphabet appartient au langage
    - Toute concaténation de mots du langage appartient au langage
    - Le mot vide ε appartient au langage
  - **Puissance** « $\sum^i$ » d'un alphabet :
    - Tous les mots formés par concaténation de i éléments de $\sum$
    - Quelque soit l'alphabet, $\sum^0 = \{ \varepsilon \}$
    - Par ex., si $\sum = \{0, 1\}$ alors :
      - $\sum^1 = \{0, 1\}$
      - $\sum^2 = \{00, 01, 10, 11\}$
      - $\sum^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

- **Fermeture / étoile de Kleene « * » :**
  - Kleene : mathématicien américain
  - Fermeture (étoile, itéré) (informellement) : tout ce qu'il est possible de construire à partir d'un alphabet / langage
  - Union de **toutes les puissances** d'un alphabet :
    - $\sum^* = \sum^0 \cup \sum^1 \cup \sum^2 \cup \sum^3 \cup \sum^4 \ldots$
  - Ou « **langage de $\sum$** », « **langage généré par $\sum$** »
  - Quelque soit $\sum$: $\varepsilon \in \sum^*$
- Attention à la distinction **symboles / mots** :
  - Si $\sum = \{0, 11\}$ alors 1.0.1 n'appartient pas à $\sum^*$
  - Si $\sum = \{a, ab\}$ alors a.ab.b.ab. n'appartient pas à $\sum^*$

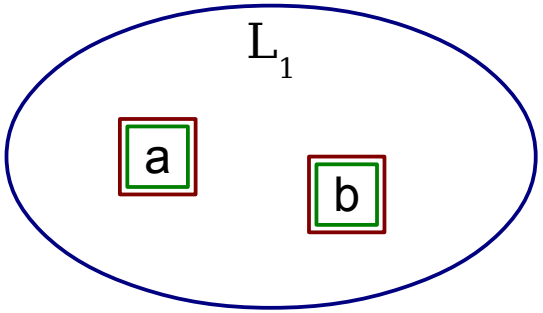
- **Langage sur un alphabet** :
  - Un langage L sur un alphabet $\sum$ est un sous-ensemble de $\sum^*$
- **Opérations** sur les langages (étant donné $\sum$) :
  - **Concaténation** : « L.M » (ou « LM ») : mot formés par concaténation d'éléments de L et de M (dans l'ordre)
    - Associative, non commutative, élément neutre ε, élément absorbant ø
  - **Puissance** : « L^i » = { $\alpha \in L^*$ t. q. $|\alpha| = i$ } (idem alphabet)
    - Remarque : $L^0 = \varepsilon$ et $L^i = L^{i-1}.L$
  - **Fermeture** (étoile, itéré) : « L^* » = $\cup_{i \geq 0} L^i$ (idem alphabet)
    - Idempotente : $(L^*)^* = L^*$
    - Quelque soient L et $\sum$: $\varepsilon \in L^*$ (idem alphabet)

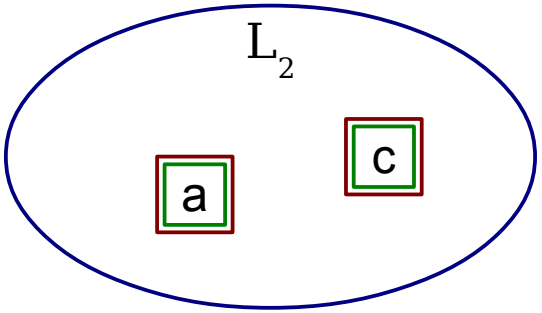
- **Union** : « L ∪ M » (ou L + M) mots issus soit de L soit de M
  - Associative, commutative, élément neutre ø
  - La concaténation est distributive par rapport à l'union
- **Intersection** : « L ∩ M » : mots qui existent dans L et dans M
  - Associative, commutative, élément neutre $\sum^*$, élément absorbant ø
- **Complémentaire** : « $\overline{L}$ » : mots de $\sum^*$ qui ne sont pas dans L
- **Fermeture** (étoile, itéré) **stricte** : : « L^+ » = $\cup_{i \geq 1} L^i$
  - Remarques : idempotente, $L^+ = L.L^*$ et $L^* = L^+ \cup \{\varepsilon\}$
- Quotient droit : « $L.M^{-1}$ » = $\{\alpha \in \sum^*, \exists \beta \in M, \alpha.\beta \in L \}$
- Quotient gauche : « $L^{-1}.M$ » = $\{\alpha \in \sum^*, \exists \beta \in L, \beta.\alpha \in M \}$
- Miroir : $\widetilde{L}$ (tilde au dessus), langage ou tous les mots sont « renversés » : l'ordre des symboles est inversé

- Quelques exemples :
  - Alphabet $\Sigma = \{ a, b, c \}$



  - Langages $L_1 = \{ a, b \}$, $L_2 = \{ a, c \}$

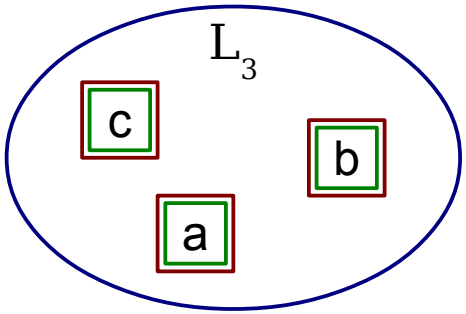




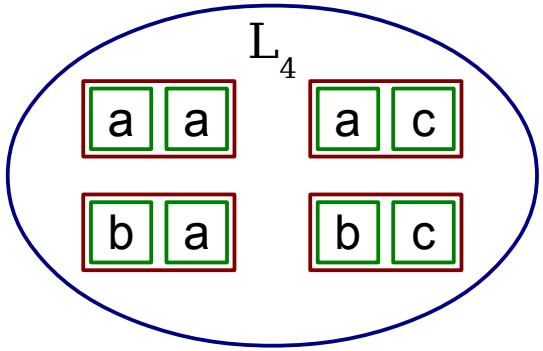
Éléments de l'alphabet

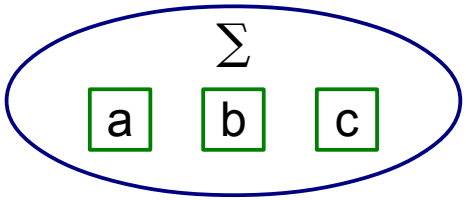
Éléments du langage

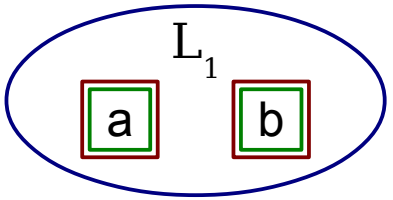
- Quelques exemples (suite) :
  - $\Sigma = \{ a, b, c \}$, $L_1 = \{ a, b \}$, $L_2 = \{ a, c \}$
  - Union : $L_3 = L_1 \cup L_2$

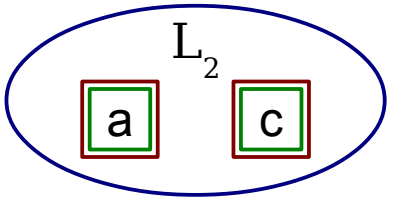


  - Concaténation : $L_4 = L_1 . L_2$





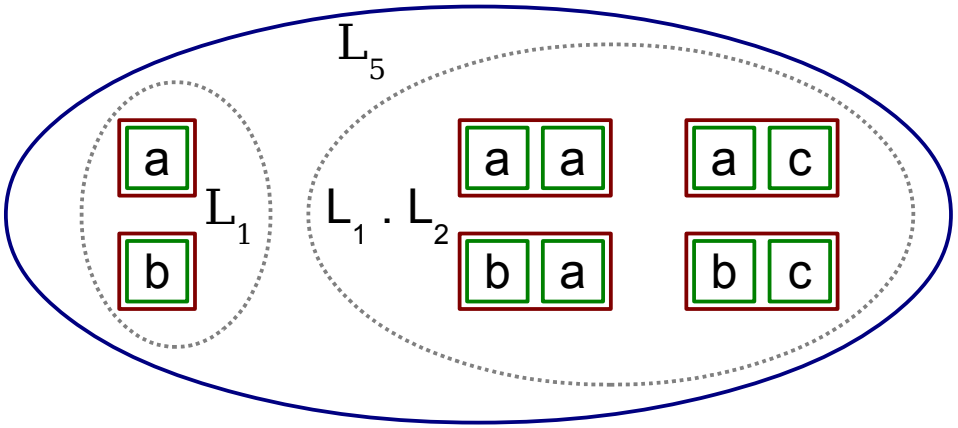




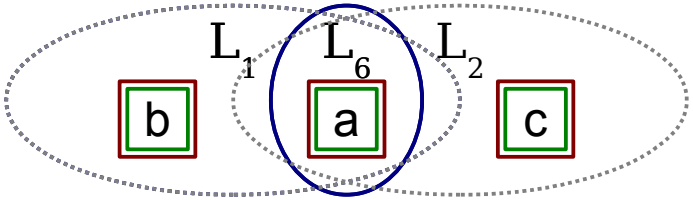
Éléments de l'alphabet

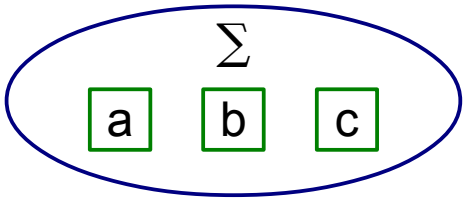
Éléments du langage

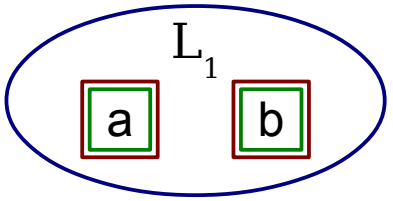
- Quelques exemples (suite) :
  - $\Sigma = \{ a, b, c \}$, $L_1 = \{ a, b \}$, $L_2 = \{ a, c \}$
  - Union et concaténation : $L_5 = L_1 \cup (L_1 . L_2)$

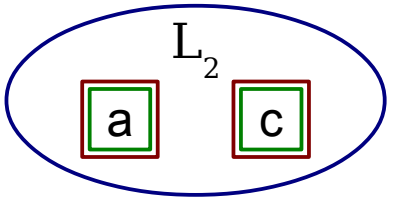


  - Intersection : $L_6 = L_1 \cap L_2$





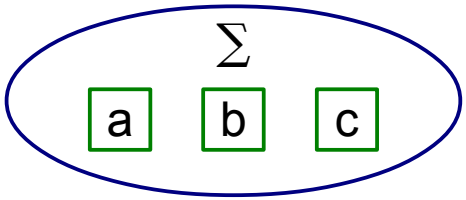


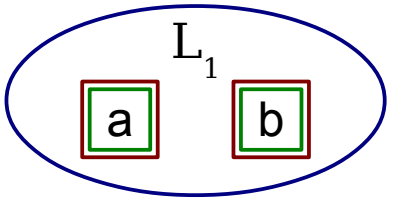


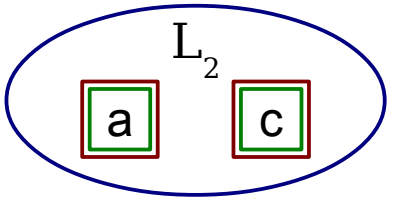
Éléments de l'alphabet

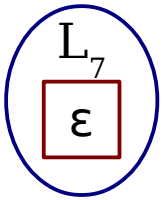
Éléments du langage

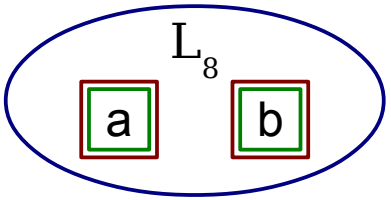
- Quelques exemples (suite) :
  - $\Sigma = \{ a, b, c \}$, $L_1 = \{ a, b \}$, $L_2 = \{ a, c \}$
  - Puissance : $L_7 = L_1^0$, $L_8 = L_1^1$, $L_9 = L_1^2$, $L_{10} = L_1^3$

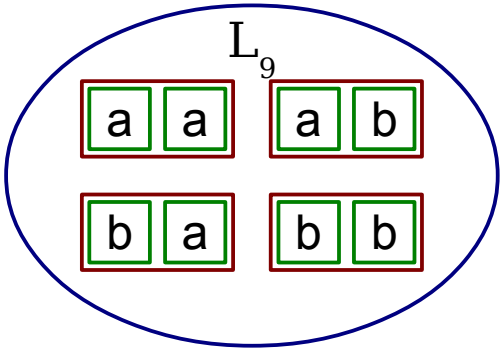


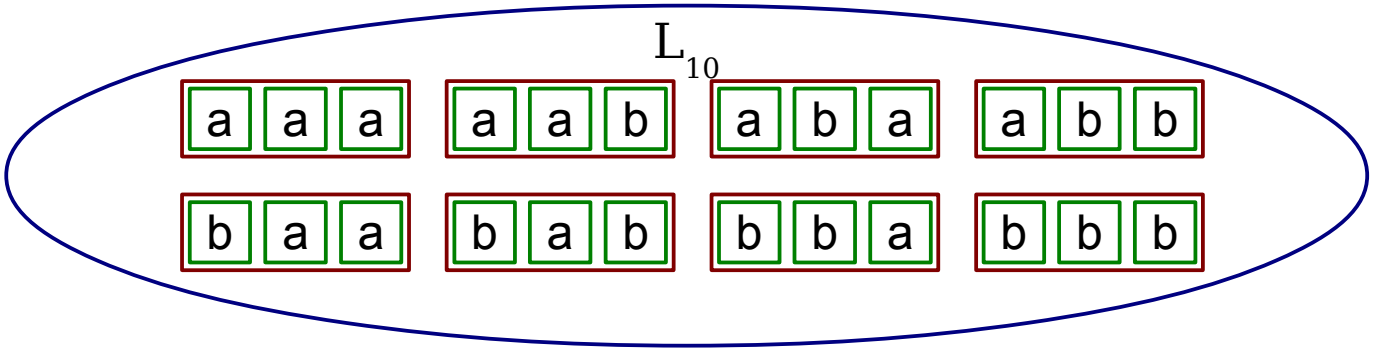








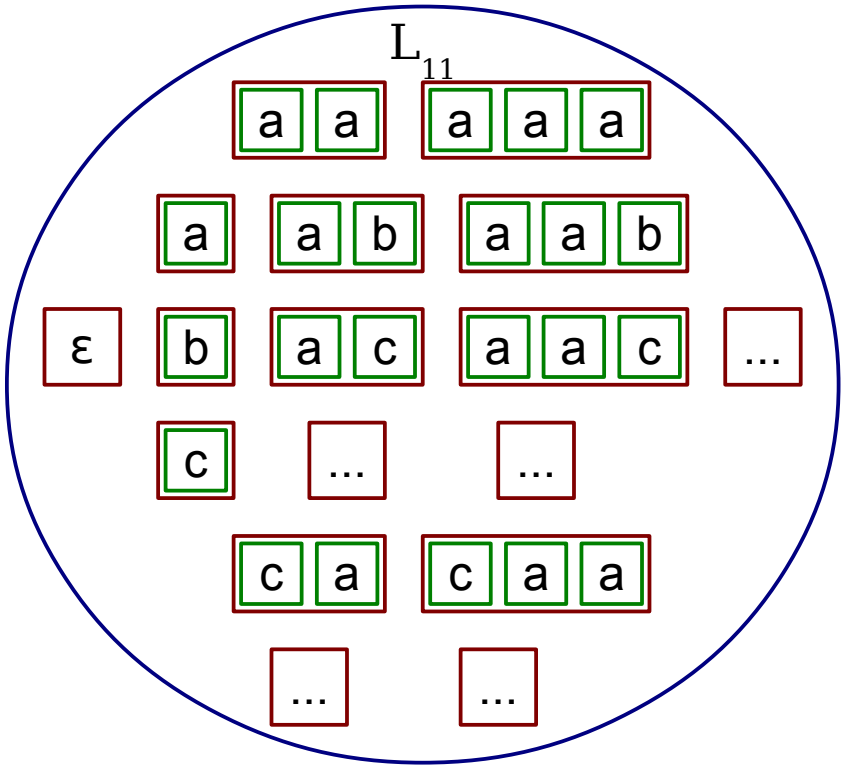


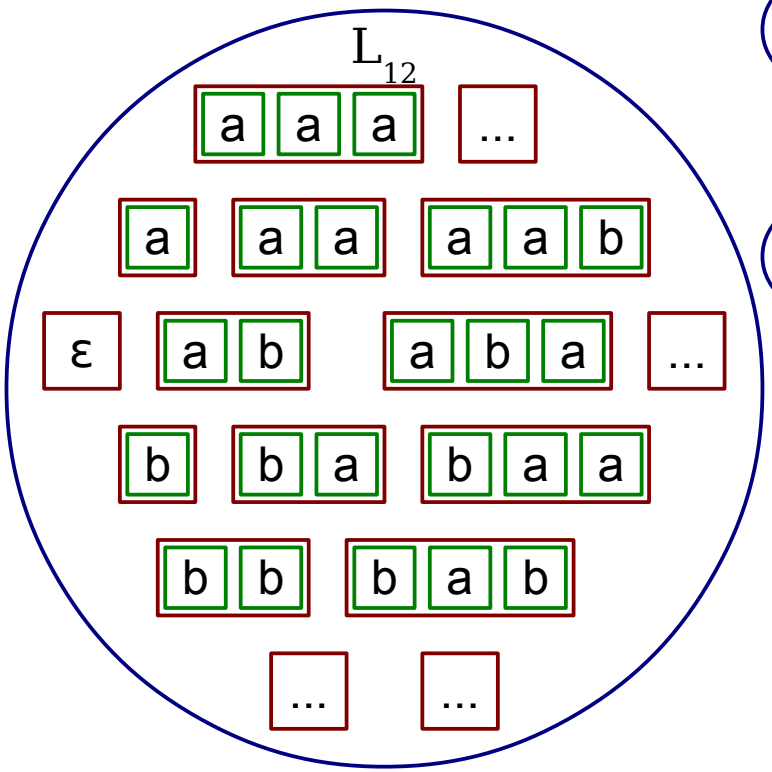


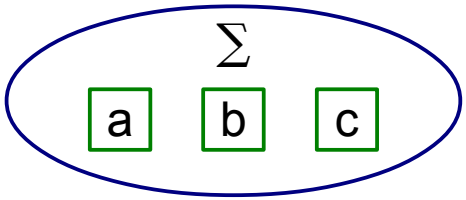
Éléments de l'alphabet

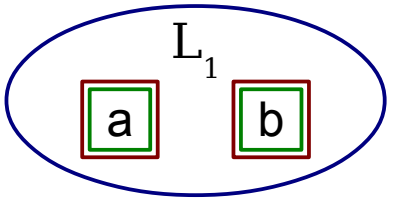
Éléments du langage

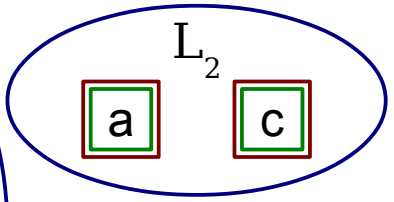
- Quelques exemples (suite) :
  - $\Sigma = \{ a, b, c \}$, $L_1 = \{ a, b \}$, $L_2 = \{ a, c \}$
  - Fermeture : $L_{11} = \Sigma^*$ ($= \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots$), $L_{12} = L_1^*$







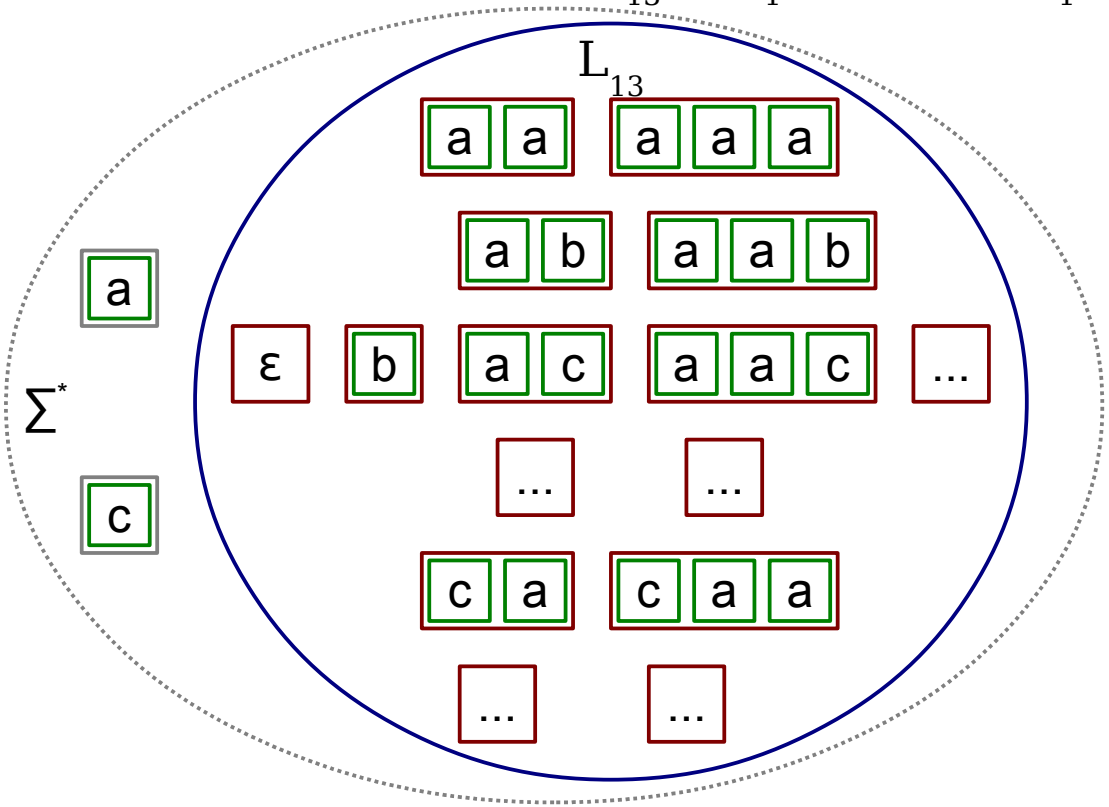


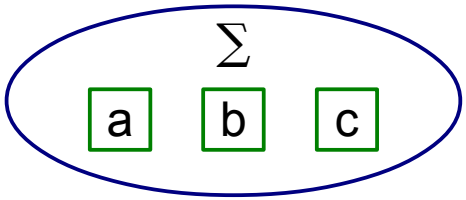


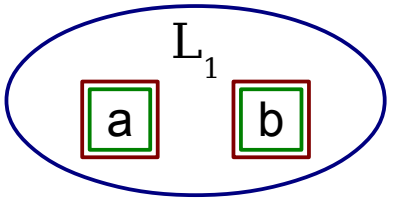
Éléments de l'alphabet

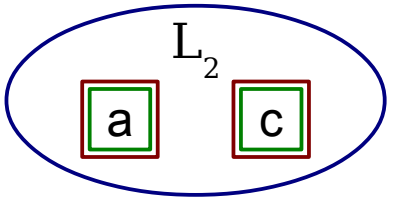
Éléments du langage

- Quelques exemples (suite) :
  - $\Sigma = \{ a, b, c \}$, $L_1 = \{ a, b \}$, $L_2 = \{ a, c \}$
  - Complémentaire : $L_{13} = \overline{L_1}$ ($\approx$ « $\Sigma^* - L_1$ »)





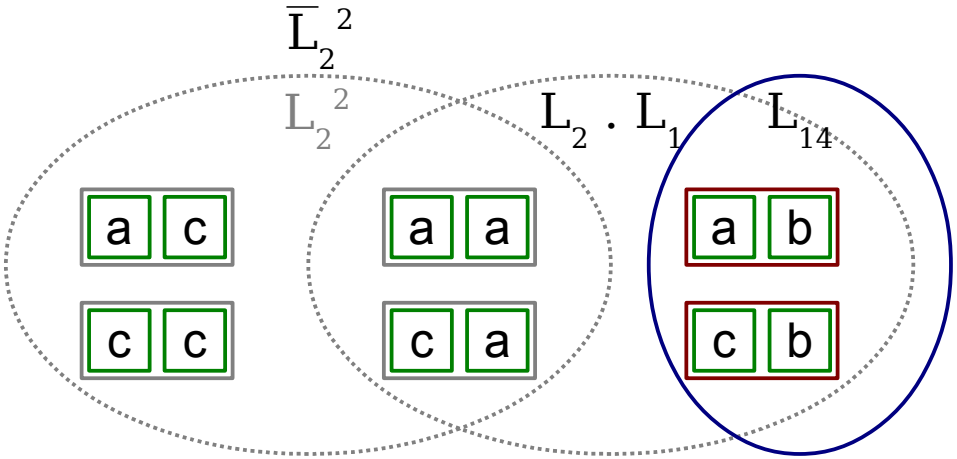


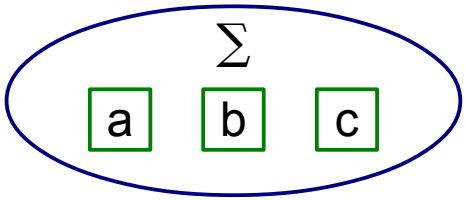


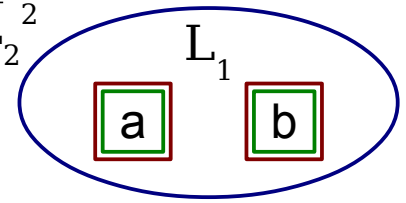
Éléments de l'alphabet

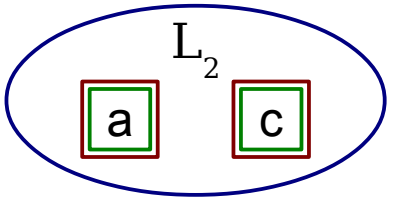
Éléments du langage

- Quelques exemples (suite) :
  - $\Sigma = \{ a, b, c \}$, $L_1 = \{ a, b \}$, $L_2 = \{ a, c \}$
  - Intersection et complémentaire : $L_{14} = (L_2 . L_1) \cap \overline{L_2}^2$









Éléments de l'alphabet

Éléments du langage

Plan

- Alphabets et langages
- Expressions régulières formelles

Théorie des langages
Expressions régulières formelles

- **Expressions régulières** :
  - ≃ expressions **rationnelles** (langage régulier ≃ rationnel)
  - Formalisées mathématiquement en 1959 (Rabin & Scott)
  - Utilisées en programmation (regexp) : **grep**, awk, Perl, Python
- Un « **langage pour définir un langage** »
  - Symboles (par priorité) : { **(,),** ***,** $^+$**, +**}
  - Une expression régulière « génère », « **accepte** », « **reconnaît** » un langage (dit régulier)
- Langage des expression régulières sur $\sum$: **Reg(∑)**
  - Reg(∑) ⊂ (∑ ∪ { **+**, *, $^+$, **(**, **)** })*
- Programmation : le + deviendra | (éviter la confusion)

- Définition **récursive** (étant donné $\sum$) :
  - Tout élément de $\sum$ est une expression régulière
  - Si r est une expression régulière, alors (r), r^+, r^* aussi
  - Si r_1 et r_2 sont des exp. régulières, alors $r_1 r_2$ et $r_1 + r_2$ aussi
- Soit l'**application L** qui associe à une **expression régulière** un **langage**, définie de la manière suivante :
  - $L : Reg(\sum) \rightarrow \sum^*$
  - $L(a) = \{ a \} \; \forall a \in \sum$, $L(\varepsilon) = \{ \varepsilon \}$ et $L(\varnothing) = \varnothing$
  - **$L(r_1 + r_2) = L(r_1) \cup L(r_2)$**
  - **$L(r_1 \, r_2) = L(r_1) . L(r_2)$**
  - **$L(r^*) = L(r)^*$ et $L(r^+) = L(r)^+$**

- Par ex., soit $\sum = \{ a, b, c \}$:
  - $L(b) = b$
  - $L(a+c) = L(a) \cup L(c) = \{a, c\}$
  - $L(ac) = L(a) . L(c) = \{a.c\}$
  - $L(c^*) = L(c)^* = \{c\}^* = \{\varepsilon, c, cc, ccc, cccc...\}$
  - $L(a+c^*) = L(a) \cup L(c^*) = \{a\} \cup \{c\}^* = \{\varepsilon, a, c, cc, ccc, cccc...\}$
  - $L((a+b)^*) = (L(a) \cup L(b))^* = \{a, b\}^*$

    $= \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab ...\}$
  - $L((ac)^+ + b) = (L(a) . L(c))^+ \cup L(b) = \{ac\}^+ \cup \{b\}$

    $= \{b, ac, acac, acacac ...\}$
  - $L(a^+ + (abc)^* + ((b+a)c)^+) = L(a)^+ \cup L(abc))^* \cup L(\{b, a\}.c)^+$

    $= \{a, aa, aaa ... \varepsilon, abc, abcabc ... bc, ac, bcbc, acac...\}$

- Quelques cas d'applications classiques :
  - Trouver des fichiers dans un dossier :
    - Tous les fichiers d'extension « .jpg » : *.jpg
  - Chercher toutes les fonctions « get... » dans un programme :
    - Expression régulière : function get[a-Z]*(
  - Extraire toutes les phrases d'un texte :
    - Expression régulière : $(\overline{.+?+!})^{*}(.+?+!)^{*}$
  - Vérifier si une entrée de programme est bien un entier :
    - Expression régulière : (-+ε){0...9)*
  - Chercher les mots au pluriel dans un texte :
    - Expression régulière : [a-z]*((e+a)ux+s)