Théorie des Langages Formels
Chapitre 1

Florence Levé

Florence.Leve@u-picardie.fr

Année 2017-2018

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 :
  - `sup(E, (E+P)/2)`,
  - pas le droit aux documents.

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

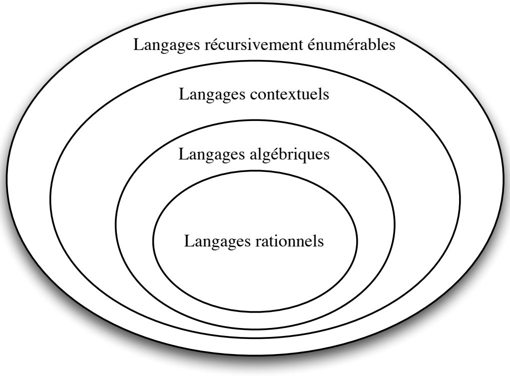
- Analyse lexicale : est-ce que mon programme utilise les mots de base du langage ?
  - utilisation d'automates dans les compilateurs.
- Analyse syntaxique : est-ce que les mots/phrases de mon programme sont correctement construits ?
  - utilisation de grammaires dans les compilateurs.
  - (problème : 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

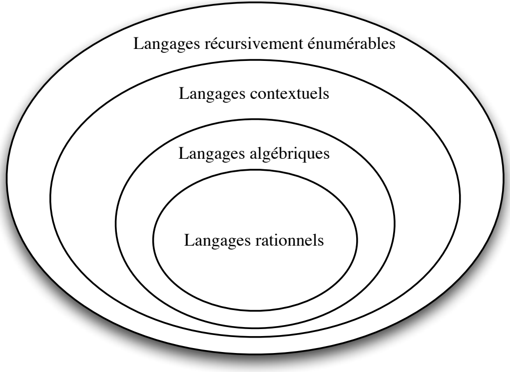
Des besoins similaires dans 3 thématiques :

- Informatique :
  - Besoin de décrire de manière 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 être 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 : premières tentatives visant à utiliser l'ordinateur pour traduire un texte
  - Noam Chomsky 1956 : hiérarchie des langages.

La Hiérarchie des langages de Chomsky

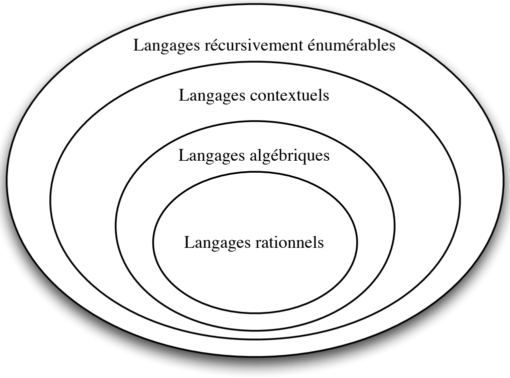


La Hiérarchie des langages de Chomsky



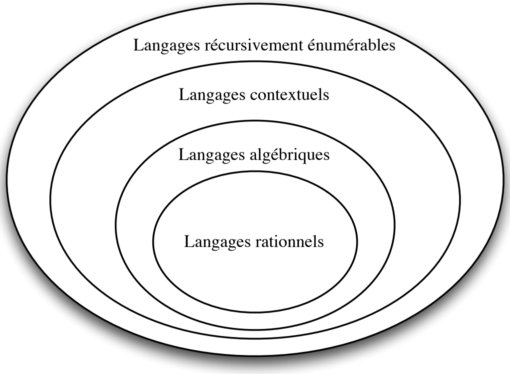
- {rationnels}
  - = {reconnaissables}
  - langages reconnus par un automate et/ou définis par une expression régulière.

La Hiérarchie des langages de Chomsky



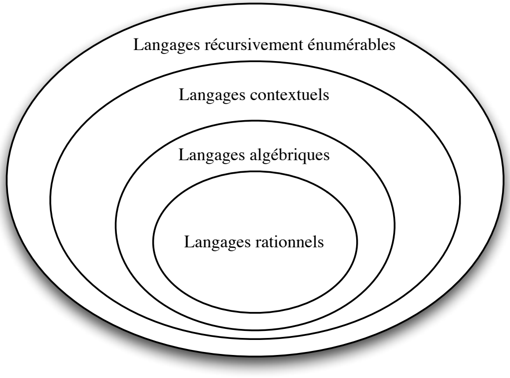
- $\subset$ {algébriques}
  - langages définis par une grammaire ou reconnus par un automate à pile.

La Hiérarchie des langages de Chomsky



- $\subset$ {contextuels}
  - langages définis par une grammaire contextuelle.

La Hiérarchie des langages de Chomsky

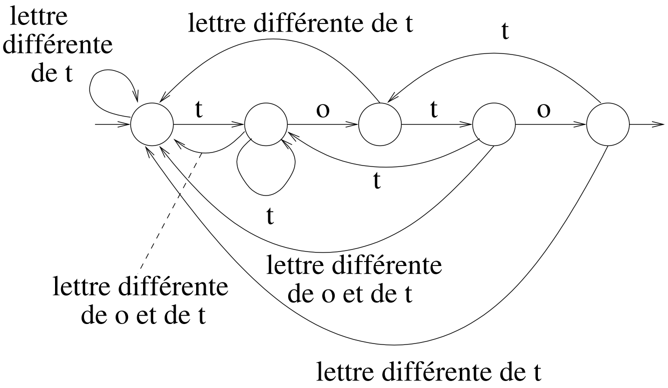


- $\subset$ {récursivement énumérables}
  - langages acceptables par une machine de Turing (permet d'étudier la décidabilité d'un problème).

Quelques utilisations

1. Recherche de motif dans un texte

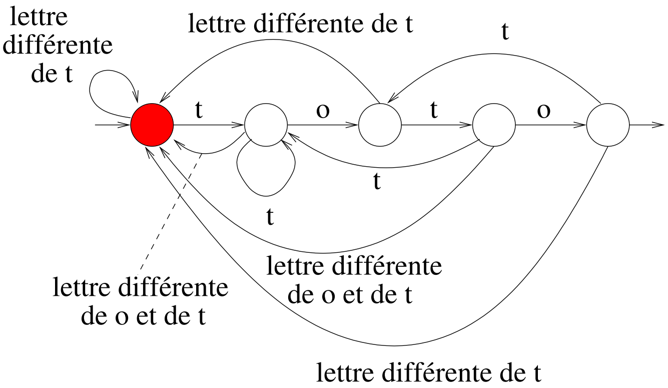
Texte en entrée :



Quelques utilisations

1. Recherche de motif dans un texte

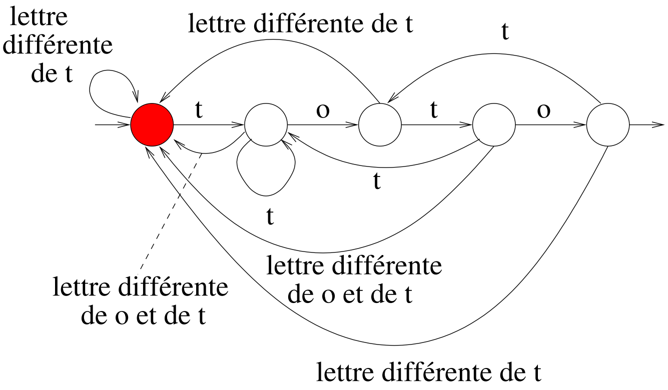
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

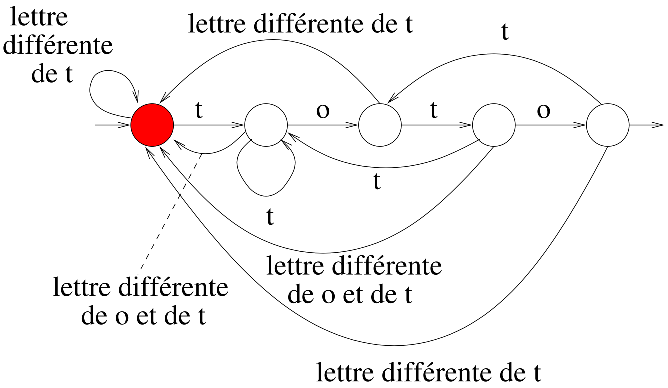
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

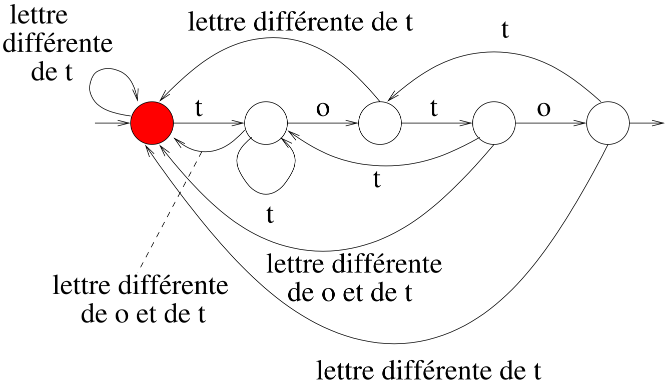
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

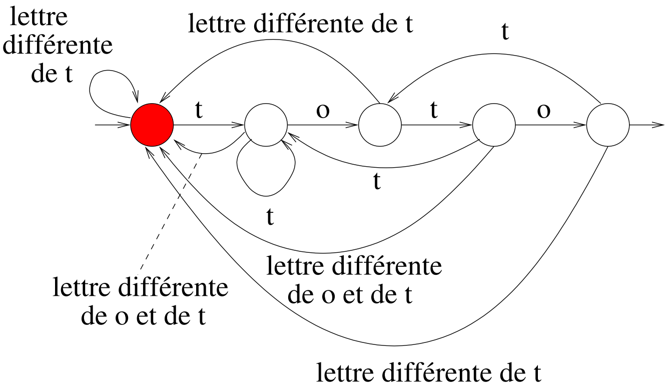
Texte en entrée : Une_histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

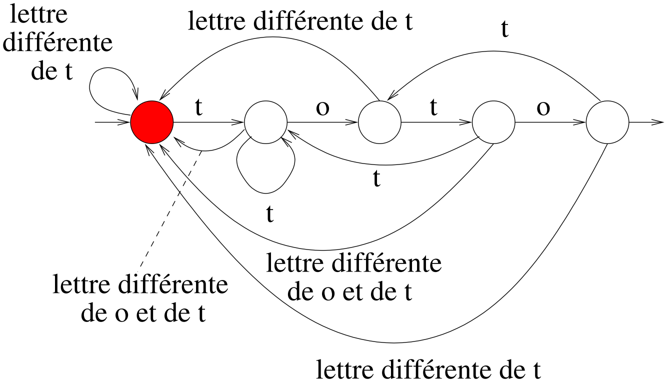
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

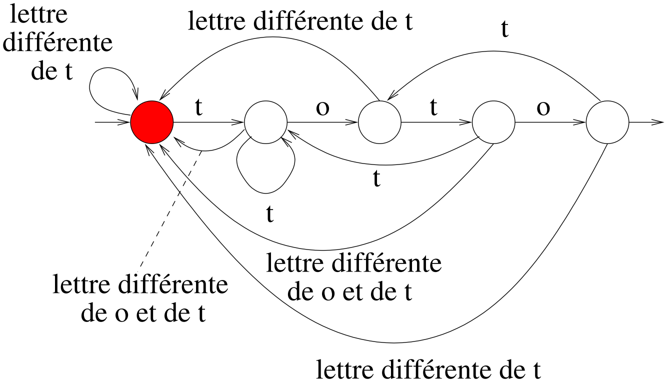
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

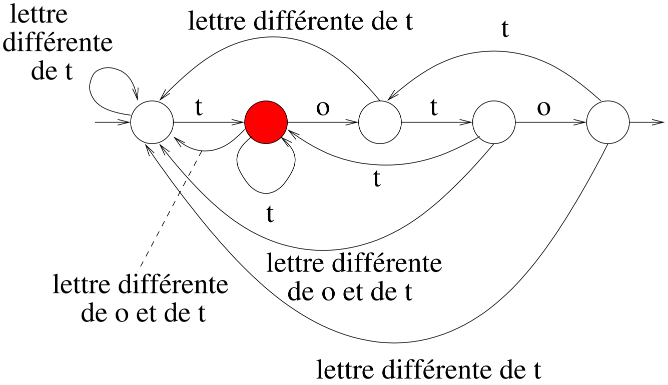
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

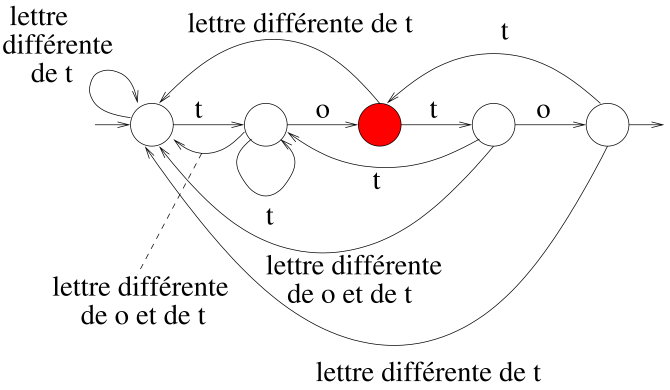
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

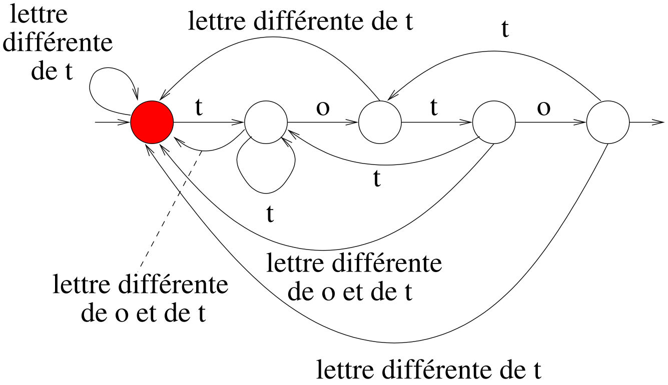
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

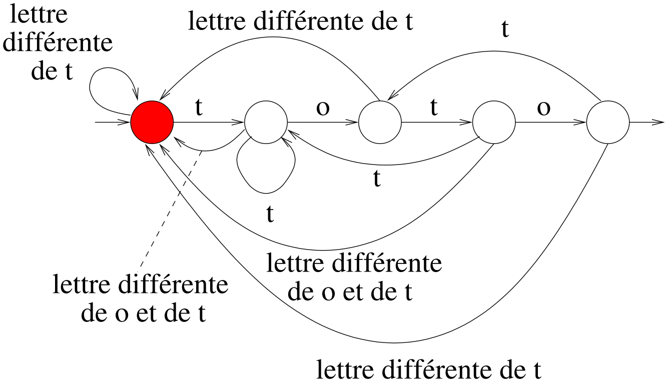
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

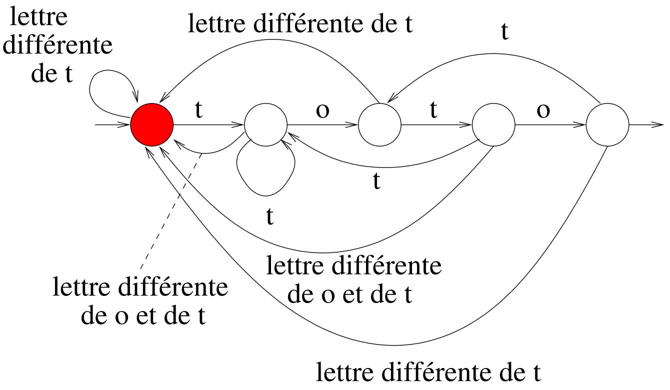
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

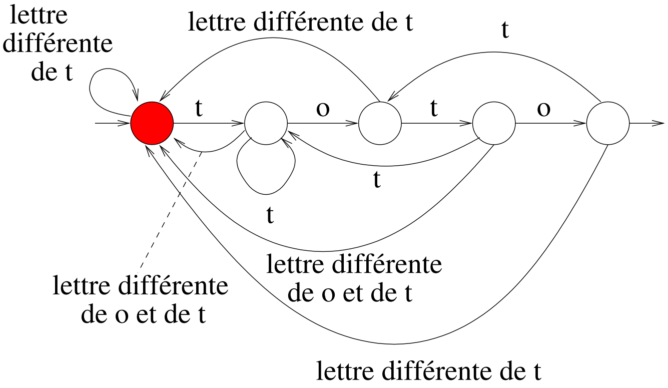
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

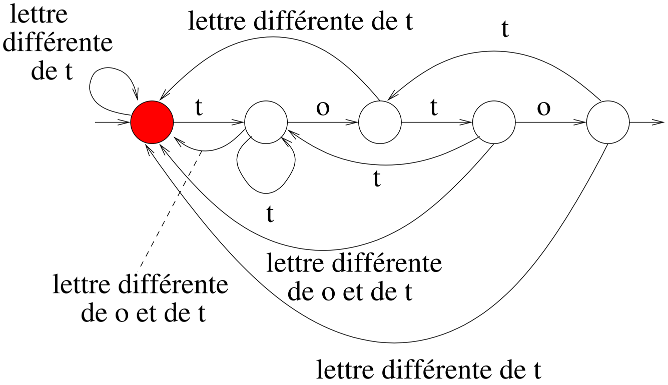
Texte en entrée : Une histoire_de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

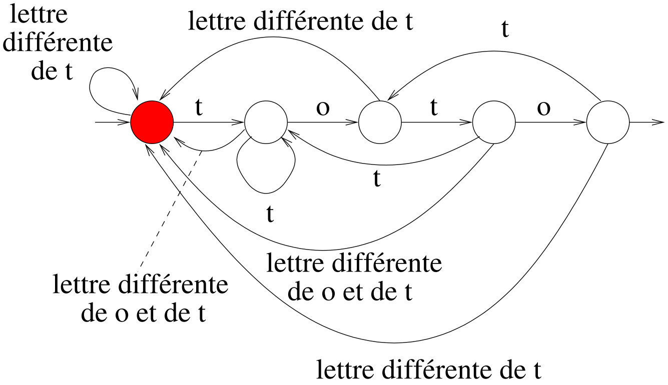
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

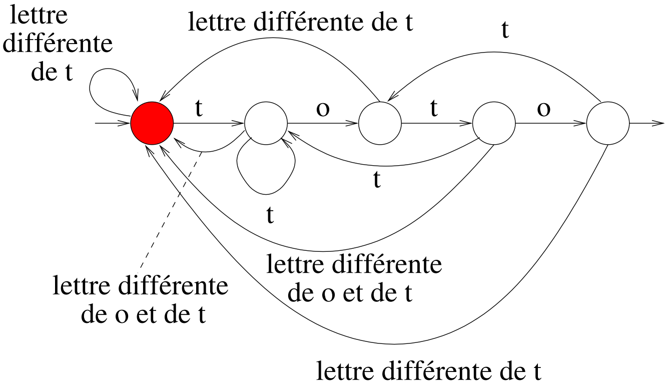
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

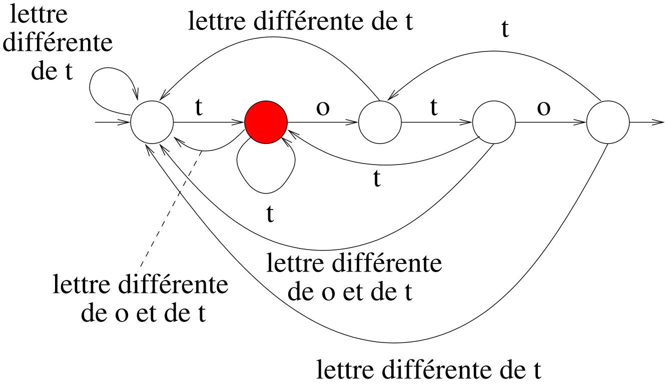
Texte en entrée : Une histoire de_toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

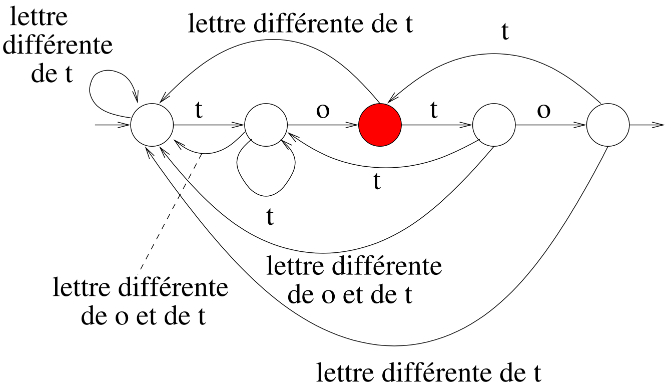
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

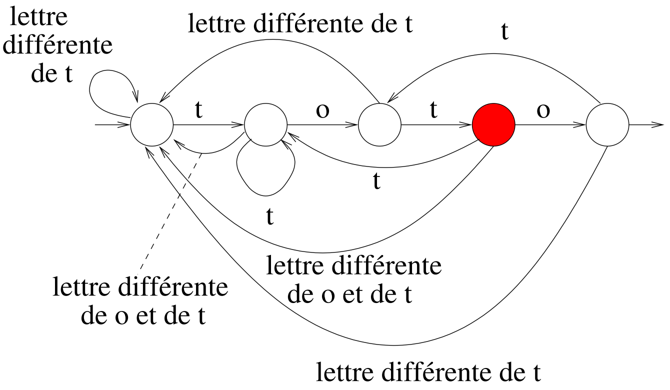
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

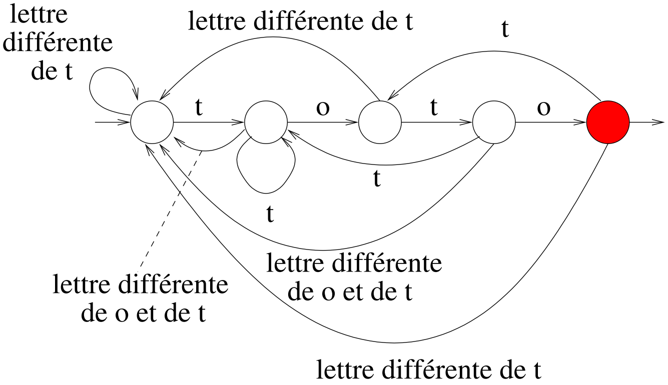
Texte en entrée : Une histoire de toto de plus



Quelques utilisations

1. Recherche de motif dans un texte

Texte en entrée : Une 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, contrôle de type, ...)

Quelques utilisations

3. REGEXP (REGular EXPressions)

- Expressions régulières
- Utilisées par les mécanismes de recherche/remplacement
  - éditeurs de texte : JEdit, emacs
  - Recherche dans des dictionnaires en ligne : dictionnaire de l'académie française (essayer par exemple `^a[a-z]*iste`)
  - 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, systèmes 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 (séquençage 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
- Grammaires

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 b.
- Notation : uv ou $u.v$
- La concaténation est aussi appelée produit de concaténation
- Exemple : $(aabac).(dab)$ vaut $aabacdab$
- Formellement, si $\left\{ \begin{array}{l} n \geq 0 \text{ et } p \geq 0 \text{ sont deux entiers,} \\ a_1, \ldots, a_n,\ b_1, \ldots, b_p \text{ sont des lettres,} \end{array} \right.$ alors

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

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

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 b.
- Notation : uv ou $u.v$
- La concaténation est aussi appelée produit de concaténation
- Exemple : $(aabac).(dab)$ vaut $aabacdab$
- Formellement, si $\left\{ \begin{array}{l} n \geq 0 \text{ et } p \geq 0 \text{ sont deux entiers,} \\ a_1, \ldots, a_n,\ b_1, \ldots, b_p \text{ sont des lettres,} \end{array} \right.$ 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.

  $a_1 \dots a_n = b_1 \dots b_p$ 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 :
  - $|\varepsilon| = 0$
  - $|uv| = |u| + |v|$

Longueur

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

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 :
  - $|\varepsilon|_a = 0$
  - $|uv|_a = |u|_a + |v|_a$
  - $|u| = \sum_{a \in A} |u|_a$

Nombre d'occurrences

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

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

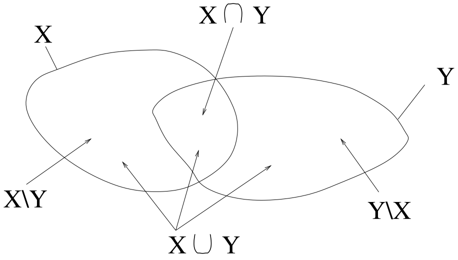
- L'ensemble constitué d'aucun élément.
- Notation : $\emptyset$.
- Exercice
  - ε :
    - le mot vide
    - d'ici à la fin de ce cours, l'expression rationnelle désignant l'ensemble $\{\varepsilon\}$
  - $\emptyset$: l'ensemble vide
  - $\{\varepsilon\}$: l'ensemble ayant comme seul élément le mot vide.
  - $\{\emptyset\}$: 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 (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 (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 (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 (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 (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 (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) $\left\{ \begin{array}{l} X \cup Y = Y \cup X, \\ X \cap Y = Y \cap X. \end{array} \right.$
- 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) $\begin{cases} X \cup (Y \cup Z) = (X \cup Y) \cup Z, \\ X \cap (Y \cap Z) = (X \cap Y) \cap Z. \end{cases}$
- 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 (3/4)

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

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

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

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

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

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 $E_1, E_2, \ldots, E_n$ ($n \geq 1$) des ensembles finis,

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

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,

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

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,

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

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 $E_1, E_2, \ldots, E_n$ ($n \geq 1$) des ensembles finis,

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

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 $E_1, E_2, \ldots, E_n$ ($\mathrm{n} \geq 1$) des ensembles finis,

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