

Théorie des langages : THL

CM 1

Uli Fahrenberg

EPITA Rennes

S5 2024

Aperçu

Langages, mots, expressions

Langages :

- de programmation
- langues naturelles
- en bio-informatique, *etc.*

Langages, mots, expressions

Langages :

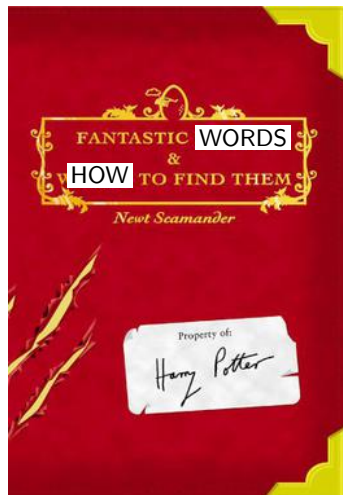
- de programmation
- langues naturelles
- en bio-informatique, *etc.*
- *qu'est-ce que* : **syntaxe**, **sémantique**

Langages, mots, expressions

Langages :

- de programmation
- langues naturelles
- en bio-informatique, *etc.*
- *qu'est-ce que* : **syntaxe**, **sémantique**

Mots :



Langages, mots, expressions

Langages :

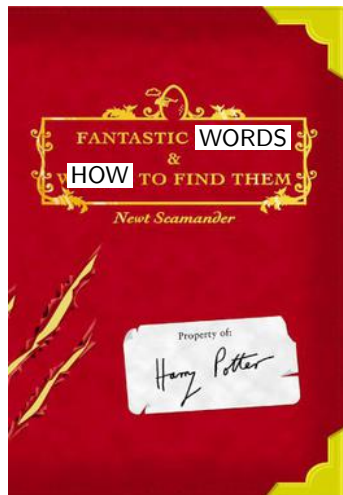
- de programmation
- langues naturelles
- en bio-informatique, *etc.*
- *qu'est-ce que* : **syntaxe**, **sémantique**

Mots :

- suite **finie** de *symboles*
- `while`, `my_var_336`,
Schallplattenabspielgerät,
`ACTAAGGT`

Expressions rationnelles :

- `[a-zA-Z][a-zA-Z0-9_]*`



Mkrtchian



Մհեր Մուշեղի Մկրտչյան

Мгер Мушегович Мкртчян

Mher Mouchérovitch Mkrtchian

Un peu (!) de précision

Symbole :

- notion axiomatique (*on s'en fout de ce que c'est*)
- $\Sigma = \{a, b, c, \dots\}$

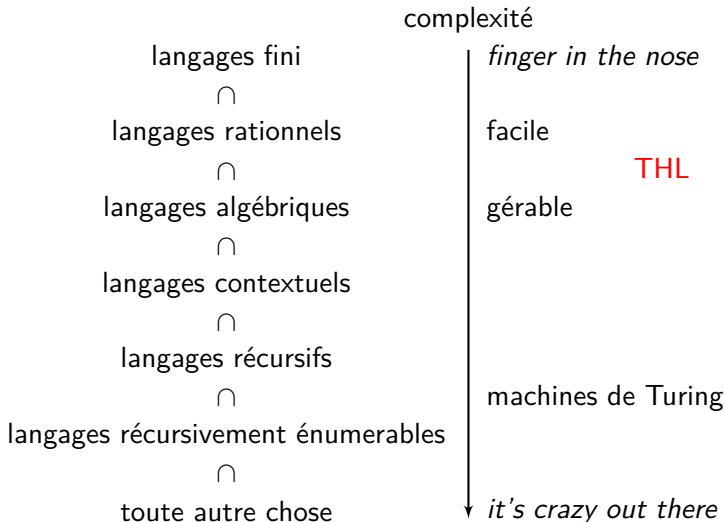
Mot :

- **suite finie** de symboles
- *a, abba, abracadabra, jenesaispasquoimaisilfautdubeurresurlepain*
- finie, mais sans limite fixe de longueur
- “Ich bin ein Berliner”, “for x in range(5)” (!)

Langage :

- **ensemble** de mots
- peut être fini (même vide !), mais normalement **infini**

Classification



Une démonstration

Définition

Un langage L est **récursivement énumérable** s'il existe un algorithme qui énumère tout les mots de L .

Exemple : $x = 2$
while true:
 if isprime(x): print(x)
 x += 1

Une démonstration

Définition

Un langage L est **récursivement énumérable** s'il existe un algorithme qui énumère tout les mots de L .

Théorème

Il existe un langage qui n'est pas récursivement énumérable.

Démonstration.

- ① L'ensemble de tous algorithmes est **dénombrable**. (Pourquoi ? Qu'est-ce que ?)
- ② Chaque algorithme n'énumère guère qu'un langage.
- ③ L'ensemble de langages n'est **pas dénombrable**. (Pourquoi ?)

... et pourquoi ?

Des applications :

- le parsing
 - expressions rationnelles
 - `grep 'a.*io.*e.*e' thlr1.txt`
- la compilation
 - analyse lexicale
 - analyse syntaxique
- la bio-informatique
 - analyse de mutations
 - « Ève mitochondriale »
- la traduction automatique

Pour en finir (la première partie)

Définition

Un algorithme A **décide** un langage donné L si, pour chaque mot w en entrée, A répond « OUI » si $w \in L$ et « NON » si $w \notin L$.

Exercice (5 mn)

- 1 Trouver un algorithme *simple* qui décide le langage de tous les mots qui **commencent par ab** :

$$L = \{ab, aba, abb, abaa, abab, abba, \dots\}$$

- 2 Trouver un algorithme *simple* qui décide le langage de tous les mots qui **se terminent par ab** :

$$L = \{ab, aab, bab, aaab, abab, baab, \dots\}$$

Infos pratiques

Le cours

- ① Langages rationnels
 - ② Automates finis
 - ③ Langages algébriques, grammaires hors-contexte
 - ④ Automates à pile
 - ⑤ Parsage LL
 - ⑥ Parsage LR
- et après, le **Tiger project** !

Le cours

- ① Langages rationnels
- ② Automates finis
 TP flex
- ③ Langages algébriques, grammaires hors-contexte
- ④ Automates à pile
- ⑤ Parsage LL
 TP LL
- ⑥ Parsage LR
 TP bison
 TP flex et bison
- et après, le **Tiger project** !

Les notes

① Langages rationnels

② Automates finis

TP flex

QCM

③ Langages algébriques, grammaires hors-contexte

④ Automates à pile

⑤ Parsage LL

TP LL

QCM

⑥ Parsage LR

TP bison

TP flex et bison

QCM

• et après, le **Tiger project** !

Le prof



Uli Fahrenberg

<https://ulifahrenberg.github.io/>
uli.fahrenberg@epita.fr

Le poly

F.Yvon, A.Demaille, **Théorie des langages**

- cours \subsetneq shuffle(chapitres 1-8)
- aujourd'hui :
 - chapitre 2, **moins** 2.3.2-5 et 2.4.4
 - chapitre 3, **moins** 3.1.3
- Moodle
- <https://ulifahrenberg.github.io/thl/>

Programme d'aujourd'hui

- ① Symboles, mots, langages
- ② L'algèbre de langages
- ③ Langages rationnels
- ④ Expressions rationnelles

Symboles, mots, langages

Symboles, mots, langages : de la précision

Soit Σ un ensemble **fini**.

- on appelle Σ un **alphabet**
- et les éléments $a, b, \dots \in \Sigma$ des **symboles**

On dénote Σ^* l'ensemble de tous les **suites finies** d'éléments de Σ .

- donc $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots = \bigcup_{n \geq 0} \Sigma^n$
- on appelle les éléments $u, v, w, \dots \in \Sigma^*$ des **mots**
- on écrit des mots $aabab$ (par exemple) au lieu de (a, a, b, a, b)

Un **langage** est un sous-ensemble $L \subseteq \Sigma^*$.

L'algèbre de mots

Il y a une opération binaire sur Σ^* :

L'algèbre de mots

Il y a une opération binaire sur Σ^* :

Définition

La **concaténation** de deux mots $a_1 \dots a_n$ et $b_1 \dots b_m$ est le mot $a_1 \dots a_n b_1 \dots b_m$.

- on utilise le symbole « . » si besoin ; sinon, rien

Voici les propriétés de la concaténation :

L'algèbre de mots

Il y a une opération binaire sur Σ^* :

Définition

La **concaténation** de deux mots $a_1 \dots a_n$ et $b_1 \dots b_m$ est le mot $a_1 \dots a_n b_1 \dots b_m$.

- on utilise le symbole « . » si besoin ; sinon, rien

Voici les propriétés de la concaténation :

Théorème

L'opération « . » est **associative** et a **le mot vide comme élément neutre** de deux côtés.

- on utilise ε pour le mot vide
- donc $u(vw) = (uv)w$, $u.\varepsilon = u$ et $\varepsilon.u = u$ pour tout $u, v, w \in \Sigma^*$
- **pas commutative**

Opérations sur langages

Opérations ensemblistes

Opérations sur langages

Opérations ensemblistes

$$L_1 \cup L_2 = \{u \in \Sigma^* \mid u \in L_1 \text{ ou } u \in L_2\}$$

$$L_1 \cap L_2 = \{u \in \Sigma^* \mid u \in L_1 \text{ et } u \in L_2\}$$

$$\overline{L} = \{u \in \Sigma^* \mid u \notin L\}$$

Opérations sur langages

Opérations ensemblistes

$$L_1 \cup L_2 = \{u \in \Sigma^* \mid u \in L_1 \text{ ou } u \in L_2\}$$

$$L_1 \cap L_2 = \{u \in \Sigma^* \mid u \in L_1 \text{ et } u \in L_2\}$$

$$\bar{L} = \{u \in \Sigma^* \mid u \notin L\}$$

Concaténation

$$L_1.L_2 = \{u_1 u_2 \mid u_1 \in L_1, u_2 \in L_2\}$$

$$L^n = L \cdots L \quad (n \text{ copies de } L)$$

Opérations sur langages

Opérations ensemblistes

$$L_1 \cup L_2 = \{u \in \Sigma^* \mid u \in L_1 \text{ ou } u \in L_2\}$$

$$L_1 \cap L_2 = \{u \in \Sigma^* \mid u \in L_1 \text{ et } u \in L_2\}$$

$$\bar{L} = \{u \in \Sigma^* \mid u \notin L\}$$

Concaténation

$$L_1.L_2 = \{u_1 u_2 \mid u_1 \in L_1, u_2 \in L_2\}$$

$$L^n = L \cdots L \quad (n \text{ copies de } L)$$

Étoile de Kleene

$$L^* = L^0 \cup L^1 \cup L^2 \cup \dots = \bigcup_{n \geq 0} L^n$$

L'algèbre de langages

Théorème

L'opération « . » sur langages est *associative* et a le langage $\{\varepsilon\}$ comme *élément neutre* de deux côtés.

- donc $L_1(L_2L_3) = (L_1L_2)L_3$, $L.\{\varepsilon\} = L$ et $\{\varepsilon\}.L = L$
- *pas commutative*
- aussi, $L.\emptyset = \emptyset$ et $\emptyset.L = \emptyset$

Théorème

$\Sigma^* = \Sigma^*$.

- (ce n'est pas une tautologie)
- aussi, $\emptyset^* = \{\varepsilon\}$, en fait $\varepsilon \in L^*$ pour chaque L

5 minutes de réflexion

Vrai ou faux ?

① $\{ab\} \cup \{ba\} = \{abba\}$

② $\{a\}^n = \{a^n\}$

③ $\{a\}^* = \{a^n \mid n \geq 0\}$

④ $\{a, b\}^n = \{a^n, b^n\}$

⑤ $\{a, b\}^3 = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$

⑥ $(L_1 \cup L_2)^2 = L_1^2 \cup L_1L_2 \cup L_2L_1 \cup L_2^2$

⑦ $L_1 \cdot (L_2 \cup L_3) = L_1L_2 \cup L_1L_3$

5 minutes de réflexion

Vrai ou faux ?

- ① $\{ab\} \cup \{ba\} = \{abba\}$ ✗
- ② $\{a\}^n = \{a^n\}$ ✓
- ③ $\{a\}^* = \{a^n \mid n \geq 0\}$ ✓
- ④ $\{a, b\}^n = \{a^n, b^n\}$ ✗
- ⑤ $\{a, b\}^3 = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$ ✓
- ⑥ $(L_1 \cup L_2)^2 = L_1^2 \cup L_1L_2 \cup L_2L_1 \cup L_2^2$ ✓
- ⑦ $L_1.(L_2 \cup L_3) = L_1L_2 \cup L_1L_3$ ✓

Longueur d'un mot

Définition

La **longueur** $|u|$ d'un mot $u \in \Sigma^*$ correspond au nombre de symboles de u .

- ($|ababa| = 5$, pas $2!$)
- donc $|\varepsilon| = 0$ et $|uv| = |u| + |v|$
- aussi, $|u| = 0$ ssi $u = \varepsilon$
- et $|u| = 1$ ssi $u \in \Sigma$

Notation

On dénote u^n la concaténation de n copies de $u \in \Sigma^*$.

- donc $(abc)^3 = abcabcabc$
- définition récursive : $u^0 = \varepsilon$ et $u^{n+1} = u u^n$
- aussi, $|u^n| = n |u|$

Préfixe, suffixe, facteur

Définition

Soit $u, v \in \Sigma^*$, alors u est un **préfixe** de v ssi il existe $w \in \Sigma^*$ tel que $uw = v$.

- des préfixes de *tomate* :

$\{ t, to, tom, toma, tomat, tomate \}$

Préfixe, suffixe, facteur

Définition

Soit $u, v \in \Sigma^*$, alors u est un **préfixe** de v ssi il existe $w \in \Sigma^*$ tel que $uw = v$.

- des préfixes de *tomate* :

$\{\epsilon, t, to, tom, toma, tomat, tomate\}$

Préfixe, suffixe, facteur

Définition

Soit $u, v \in \Sigma^*$, alors u est un **préfixe** de v ssi il existe $w \in \Sigma^*$ tel que $uw = v$.

- des préfixes de *tomate* :

$\{\epsilon, t, to, tom, toma, tomat, tomate\}$

Définition

Soit $u, v \in \Sigma^*$, alors

- u est un **suffixe** de v ssi $\exists w \in \Sigma^* : wu = v$;
- u est un **facteur** de v ssi $\exists w_1, w_2 \in \Sigma^* : w_1 u w_2 = v$.

Préfixe, suffixe, facteur : suite

Pour un langage $L \subseteq \Sigma^*$ on note le **langage de préfixes** de L par

$$\text{Pref}(L) = \{u \in \Sigma^* \mid \exists v \in L : u \text{ préfixe de } v\}$$

- donc $\text{Pref}(\{tomate\}) = \{\varepsilon, t, to, tom, toma, tomat, tomate\}$
- même chose pour $\text{Suff}(L)$ et $\text{Fact}(L)$

Vrai ou faux ? (5 mn)

- 1 $\text{Fact}(L) = \text{Pref}(L) \cup \text{Suff}(L)$
- 2 $\text{Pref}(\text{Pref}(L)) = \text{Pref}(L)$
- 3 $\text{Pref}(\text{Fact}(L)) = \text{Pref}(L)$
- 4 $\text{Pref}(\text{Fact}(L)) = \text{Fact}(L)$
- 5 $\text{Pref}(\text{Suff}(L)) = \text{Suff}(\text{Pref}(L)) = \text{Fact}(L)$

Préfixe, suffixe, facteur : suite

Pour un langage $L \subseteq \Sigma^*$ on note le **langage de préfixes** de L par

$$\text{Pref}(L) = \{u \in \Sigma^* \mid \exists v \in L : u \text{ préfixe de } v\}$$

- donc $\text{Pref}(\{tomate\}) = \{\varepsilon, t, to, tom, toma, tomat, tomate\}$
- même chose pour $\text{Suff}(L)$ et $\text{Fact}(L)$

Vrai ou faux ? (5 mn)

- | | |
|--|---|
| ① $\text{Fact}(L) = \text{Pref}(L) \cup \text{Suff}(L)$ | ✗ |
| ② $\text{Pref}(\text{Pref}(L)) = \text{Pref}(L)$ | ✓ |
| ③ $\text{Pref}(\text{Fact}(L)) = \text{Pref}(L)$ | ✗ |
| ④ $\text{Pref}(\text{Fact}(L)) = \text{Fact}(L)$ | ✓ |
| ⑤ $\text{Pref}(\text{Suff}(L)) = \text{Suff}(\text{Pref}(L)) = \text{Fact}(L)$ | ✓ |

Exercice

Redéfinissez chacun des langages suivants, sur alphabet $\Sigma = \{a, b\}$, en n'utilisant **que des ensembles finis et des opérations \cup , $.$ et $*$** :

- ① $\text{Pref}(\{a\}\{b\}^*)$
- ② $\text{Suff}(\{a\}\{b\}^*)$
- ③ $\{a\}\{b\}^*\{a\}^* \cap \{a\}^*\{b\}^*\{a\}$
- ④ $\overline{\{a\}^*}$

Exercice

Redéfinissez chacun des langages suivants, sur alphabet $\Sigma = \{a, b\}$, en n'utilisant **que des ensembles finis et des opérations \cup , $.$ et $*$** :

- ① $\text{Pref}(\{a\}\{b\}^*) = \{\varepsilon, a\} \cup \{a\}\{b\}^*$
- ② $\text{Suff}(\{a\}\{b\}^*) = \{a\}\{b\}^* \cup \{b\}^*$
- ③ $\{a\}\{b\}^*\{a\}^* \cap \{a\}^*\{b\}^*\{a\} = \{a\}\{b\}^*\{a\} \cup \{a\}\{a\}^*$
- ④ $\overline{\{a\}^*} = \{a, b\}^*\{b\}\{a, b\}^*$

Langages rationnels

5 minutes de réflexion

Vrai ou faux ?

- ❶ $\{a\}^n = \{a^n\}$
- ❷ $\{a, b\}^n = \{a^n, b^n\}$
- ❸ L^* est un ensemble infini pour tout $L \subseteq \Sigma^*$
- ❹ pour tout $L \subseteq \Sigma^*$ et $n \in \mathbb{N}$, l'ensemble $\{u \in L \mid |u| \leq n\}$ est un ensemble fini
- ❺ $\{a, b\}^* = \{a\}^* \{b\}^*$
- ❻ $\{a, b\}^* = (\{a\}^* \{b\}^*)^* \{a\}^*$
- ❼ $(L_1 \cup L_2)^* = (L_1^* L_2)^* L_1^*$

5 minutes de réflexion

Vrai ou faux ?

- ① $\{a\}^n = \{a^n\}$ ✓
- ② $\{a, b\}^n = \{a^n, b^n\}$ ✗
- ③ L^* est un ensemble infini pour tout $L \subseteq \Sigma^*$ ✗
- ④ pour tout $L \subseteq \Sigma^*$ et $n \in \mathbb{N}$, l'ensemble $\{u \in L \mid |u| \leq n\}$ est un ensemble fini ✓
- ⑤ $\{a, b\}^* = \{a\}^* \{b\}^*$ ✗
- ⑥ $\{a, b\}^* = (\{a\}^* \{b\}^*)^* \{a\}^*$ ✓
- ⑦ $(L_1 \cup L_2)^* = (L_1^* L_2)^* L_1^*$ ✓

Opérations rationnelles

Soit Σ un alphabet, on travaille avec des langages dans $\mathcal{P}(\Sigma^*)$.

On a vu dans l'exo que les opérations \cup , \cdot et $*$ sont bien spéciales.

- $\text{Pref}(\{a\}\{b\}^*) = \{\varepsilon, a\} \cup \{a\}\{b\}^*$
- $\text{Suff}(\{a\}\{b\}^*) = \{a\}\{b\}^* \cup \{b\}^*$
- etc.

Définition

Les **opérations rationnelles** dans $\mathcal{P}(\Sigma^*)$ sont \cup , \cdot et $*$.

- donc union, concaténation et étoile de Kleene

Théorème (pour plus tard) : Toutes les autres opérations sont exprimables par \cup , \cdot et $*$.

Langages rationnels

Définition (3.1)

Les **langages rationnels** sur Σ sont définis inductivement comme suite :

- 1 \emptyset et $\{\varepsilon\}$ sont des langages rationnels
- 2 pour tout $a \in \Sigma$, $\{a\}$ est un langage rationnel
- 3 si L_1 et L_2 sont des langages rationnels, alors $L_1 \cup L_2$, $L_1.L_2$ et L_1^* le sont également

- $\{\varepsilon\} = \emptyset^* \Rightarrow$ on peut enlever $\{\varepsilon\}$ de la définition

Lemme

L est rationnel si et seulement si

- $L = \emptyset$ ou $L = \{a\}$ pour un $a \in \Sigma$ ou
- $L = L_1 \cup L_2$, $L = L_1 L_2$ ou $L = L_1^*$ pour L_1 et L_2 rationnels.

(En quoi ce lemme est-il différent de la définition ?)

Rationalité

Théorème

Si L_1 et L_2 sont des langages rationnels, alors $L_1 \cap L_2$, \bar{L}_1 , $\text{Pref}(L_1)$, $\text{Suff}(L_1)$ et $\text{Fact}(L_1)$ le sont aussi.

- pour la démonstration faut attendre un peu

5 minutes de réflexion

Rationnel ou pas rationnel, sur alphabet $\Sigma = \{a, b, c\}$?

- ① $\{a, b, abcba\}$
- ② $\{a^n \mid n \geq 0\}$
- ③ $\{w \in \Sigma^* \mid w \text{ contient au moins trois } a\}$
- ④ $\{w \in \Sigma^* \mid |w| \geq 5\}$
- ⑤ $\{a^{2n} \mid n \geq 0\}$
- ⑥ $\{a^{n^2} \mid n \geq 0\}$
- ⑦ $\{a^m b^n \mid m, n \geq 0\}$
- ⑧ $\{a^n b^n \mid n \geq 0\}$

5 minutes de réflexion

Rationnel ou pas rationnel, sur alphabet $\Sigma = \{a, b, c\}$?

- | | | |
|---|--|---|
| ① | $\{a, b, abcba\}$ | ✓ |
| ② | $\{a^n \mid n \geq 0\}$ | ✓ |
| ③ | $\{w \in \Sigma^* \mid w \text{ contient au moins trois } a\}$ | ✓ |
| ④ | $\{w \in \Sigma^* \mid w \geq 5\}$ | ✓ |
| ⑤ | $\{a^{2n} \mid n \geq 0\}$ | ✓ |
| ⑥ | $\{a^{n^2} \mid n \geq 0\}$ | ✗ |
| ⑦ | $\{a^m b^n \mid m, n \geq 0\}$ | ✓ |
| ⑧ | $\{a^n b^n \mid n \geq 0\}$ | ✗ |

Expressions rationnelles

Expressions rationnelles

Une notation pratique pour des langages rationnels :

Définition (3.2)

Les **expressions rationnelles** sur Σ sont définis inductivement comme suite :

- ① \emptyset et ε sont des expressions rationnelles
- ② pour tout $a \in \Sigma$, a est une expression rationnelle
- ③ si e_1 et e_2 sont des expressions rationnelles, alors $e_1 + e_2$, $e_1 \cdot e_2$ et e_1^* le sont également

Expressions rationnelles

Une notation pratique pour des langages rationnels :

Définition (3.1, *recall*)

Les **langages rationnels** sur Σ sont définis inductivement comme suite :

- ① \emptyset et $\{\varepsilon\}$ sont des langages rationnels
 - ② pour tout $a \in \Sigma$, $\{a\}$ est un langage rationnel
 - ③ si L_1 et L_2 sont des langages rationnels, alors $L_1 \cup L_2$, $L_1.L_2$ et L_1^* le sont également
- presque la même chose ! **mais**
 - 3.1 introduit une classe de sous-ensembles de Σ^* ,
 - 3.2 définit des expressions syntaxiques

Expressions rationnelles

Une notation pratique pour des langages rationnels :

Définition (3.2)

Les **expressions rationnelles** sur Σ sont définis inductivement comme suite :

- ① \emptyset et ε sont des expressions rationnelles
- ② pour tout $a \in \Sigma$, a est une expression rationnelle
- ③ si e_1 et e_2 sont des expressions rationnelles, alors $e_1 + e_2$, $e_1 \cdot e_2$ et e_1^* le sont également

- presque la même chose ! **mais**
- 3.1 introduit une classe de sous-ensembles de Σ^* ,
- 3.2 définit des expressions syntaxiques

On va relier les deux en donnant une **sémantique** aux expressions rationnelles.

Sémantique

Définition

Le **langage dénoté** par une expression rationnelle e sur Σ est $L(e) \subseteq \Sigma^*$ définit inductivement comme suite :

- ① $L(\emptyset) = \emptyset, L(\varepsilon) = \{\varepsilon\}$
- ② $L(a) = \{a\}$ pour tout $a \in \Sigma$
- ③ $L(e_1 + e_2) = L(e_1) \cup L(e_2), L(e_1 \cdot e_2) = L(e_1) \cdot L(e_2),$
 $L(e^*) = (L(e))^*$

Théorème

$L \subseteq \Sigma^*$ est rationnel ssi il existe une expression rationnelle e telle que $L = L(e)$.

Démonstration.

Par **induction structurelle** (sur tableau).

La démonstration (sur tableau)

Les **langages rationnels** sur Σ :

- ① \emptyset et $\{\varepsilon\}$ sont des langages rationnels
- ② pour tout $a \in \Sigma$, $\{a\}$ est un langage rationnel
- ③ L_1 et L_2 langages rationnels $\Rightarrow L_1 \cup L_2$, $L_1.L_2$ et L_1^* aussi

Les **expressions rationnelles** sur Σ :

- ① \emptyset et ε sont des expressions rationnelles
- ② pour tout $a \in \Sigma$, a est une expression rationnelle
- ③ e_1 et e_2 expressions rationnelles $\Rightarrow e_1 + e_2$, $e_1.e_2$ et e_1^* aussi

Le **langage dénoté** par e :

- ① $L(\emptyset) = \emptyset$, $L(\varepsilon) = \{\varepsilon\}$
- ② $L(a) = \{a\}$ pour tout $a \in \Sigma$
- ③ $L(e_1 + e_2) = L(e_1) \cup L(e_2)$, $L(e_1.e_2) = L(e_1).L(e_2)$,
 $L(e^*) = (L(e))^*$

Exemples

uli@sibelius: ~/THLR

```
uli@sibelius:~/THLR$ grep Chapitre theorie-des-langages.txt
```

Exemples

uli@sibelius: ~/THLR

```
uli@sibelius:~/THLR$ grep Chapitre theorie-des-langages.txt
```

Chapitre 1

Chapitre 2

Chapitre 3

Chapitre 4

Chapitre 5

Chapitre 6

Chapitre 7

Chapitre 8

Chapitre 9

Chapitre 10

Chapitre rédigé par Pierre Senellart.

Chapitre 11

Chapitre rédigé par Pierre Senellart.

Chapitre 12

Chapitre 13

Chapitre 14

Chapitre 15

Chapitre 16

Chapitre 17

Chapitre 18

```
uli@sibelius:~/THLR$
```


Exemples

uli@sibelius: ~/THLR

```
uli@sibelius:~/THLR$ grep 'Chapitre [0-9]\+' theorie-des-langages.txt
```

Exemples

uli@sibelius: ~/THLR

```
uli@sibelius:~/THLR$ grep 'Chapitre [0-9]\+' theorie-des-langages.txt
```

Chapitre 1

Chapitre 2

Chapitre 3

Chapitre 4

Chapitre 5

Chapitre 6

Chapitre 7

Chapitre 8

Chapitre 9

Chapitre 10

Chapitre 11

Chapitre 12

Chapitre 13

Chapitre 14

Chapitre 15

Chapitre 16

Chapitre 17

Chapitre 18

```
uli@sibelius:~/THLR$
```

Encore une démonstration

Théorème

Si L est un langage rationnel, alors $\text{Pref}(L)$ l'est aussi.

Démonstration.

Encore une démonstration

Théorème

Si L est un langage rationnel, alors $\text{Pref}(L)$ l'est aussi.

Démonstration.

- 1 Soit e une expression rationnelle telle que $L = L(e)$.

Encore une démonstration

Théorème

Si L est un langage rationnel, alors $\text{Pref}(L)$ l'est aussi.

Démonstration.

- 1 Soit e une expression rationnelle telle que $L = L(e)$.
- 2 Nous construirons une expression rationnelle $\text{pref}(e)$ telle que $L(\text{pref}(e)) = \text{Pref}(L)$.

Encore une démonstration

Théorème

Si L est un langage rationnel, alors $\text{Pref}(L)$ l'est aussi.

Démonstration.

- 1 Soit e une expression rationnelle telle que $L = L(e)$.
- 2 Nous construirons une expression rationnelle $\text{pref}(e)$ telle que $L(\text{pref}(e)) = \text{Pref}(L)$.
- 3 ... par induction structurelle :

Encore une démonstration

Théorème

Si L est un langage rationnel, alors $\text{Pref}(L)$ l'est aussi.

Démonstration.

- ① Soit e une expression rationnelle telle que $L = L(e)$.
- ② Nous construirons une expression rationnelle $\text{pref}(e)$ telle que $L(\text{pref}(e)) = \text{Pref}(L)$.
- ③ ... par induction structurelle :
- ④ $\text{pref}(\emptyset) = \quad$, $\text{pref}(\varepsilon) = \quad$, $\text{pref}(a) = \quad$

Encore une démonstration

Théorème

Si L est un langage rationnel, alors $\text{Pref}(L)$ l'est aussi.

Démonstration.

- 1 Soit e une expression rationnelle telle que $L = L(e)$.
- 2 Nous construirons une expression rationnelle $\text{pref}(e)$ telle que $L(\text{pref}(e)) = \text{Pref}(L)$.
- 3 ... par induction structurelle :
- 4 $\text{pref}(\emptyset) = \emptyset$, $\text{pref}(\varepsilon) = \varepsilon$, $\text{pref}(a) = a + \varepsilon$

Encore une démonstration

Théorème

Si L est un langage rationnel, alors $\text{Pref}(L)$ l'est aussi.

Démonstration.

- ① Soit e une expression rationnelle telle que $L = L(e)$.
- ② Nous construirons une expression rationnelle $\text{pref}(e)$ telle que $L(\text{pref}(e)) = \text{Pref}(L)$.
- ③ ... par induction structurelle :
- ④ $\text{pref}(\emptyset) = \emptyset$, $\text{pref}(\varepsilon) = \varepsilon$, $\text{pref}(a) = a + \varepsilon$
- ⑤ $\text{pref}(e_1 + e_2) =$
 $\text{pref}(e_1 e_2) =$
 $\text{pref}(e^*) =$

Encore une démonstration

Théorème

Si L est un langage rationnel, alors $\text{Pref}(L)$ l'est aussi.

Démonstration.

- ① Soit e une expression rationnelle telle que $L = L(e)$.
- ② Nous construirons une expression rationnelle $\text{pref}(e)$ telle que $L(\text{pref}(e)) = \text{Pref}(L)$.
- ③ ... par induction structurelle :
- ④ $\text{pref}(\emptyset) = \emptyset$, $\text{pref}(\varepsilon) = \varepsilon$, $\text{pref}(a) = a + \varepsilon$
- ⑤ $\text{pref}(e_1 + e_2) = \text{pref}(e_1) + \text{pref}(e_2)$
 $\text{pref}(e_1 e_2) =$
 $\text{pref}(e^*) =$

Encore une démonstration

Théorème

Si L est un langage rationnel, alors $\text{Pref}(L)$ l'est aussi.

Démonstration.

- ① Soit e une expression rationnelle telle que $L = L(e)$.
- ② Nous construirons une expression rationnelle $\text{pref}(e)$ telle que $L(\text{pref}(e)) = \text{Pref}(L)$.
- ③ ... par induction structurelle :
- ④ $\text{pref}(\emptyset) = \emptyset$, $\text{pref}(\varepsilon) = \varepsilon$, $\text{pref}(a) = a + \varepsilon$
- ⑤ $\text{pref}(e_1 + e_2) = \text{pref}(e_1) + \text{pref}(e_2)$
 $\text{pref}(e_1 e_2) = \text{pref}(e_1) + e_1 \text{pref}(e_2)$
 $\text{pref}(e^*) =$

Encore une démonstration

Théorème

Si L est un langage rationnel, alors $\text{Pref}(L)$ l'est aussi.

Démonstration.

- ① Soit e une expression rationnelle telle que $L = L(e)$.
- ② Nous construirons une expression rationnelle $\text{pref}(e)$ telle que $L(\text{pref}(e)) = \text{Pref}(L)$.
- ③ ... par induction structurelle :
- ④ $\text{pref}(\emptyset) = \emptyset$, $\text{pref}(\varepsilon) = \varepsilon$, $\text{pref}(a) = a + \varepsilon$
- ⑤ $\text{pref}(e_1 + e_2) = \text{pref}(e_1) + \text{pref}(e_2)$
 $\text{pref}(e_1 e_2) = \text{pref}(e_1) + e_1 \text{pref}(e_2)$
 $\text{pref}(e^*) = e^* \text{pref}(e)$ (voir tableau)

Encore une démonstration

Théorème

Si L est un langage rationnel, alors $\text{Pref}(L)$ l'est aussi.

Démonstration.

- ① Soit e une expression rationnelle telle que $L = L(e)$.
- ② Nous construirons une expression rationnelle $\text{pref}(e)$ telle que $L(\text{pref}(e)) = \text{Pref}(L)$.
- ③ ... par induction structurelle :
- ④ $\text{pref}(\emptyset) = \emptyset$, $\text{pref}(\varepsilon) = \varepsilon$, $\text{pref}(a) = a + \varepsilon$
- ⑤ $\text{pref}(e_1 + e_2) = \text{pref}(e_1) + \text{pref}(e_2)$
 $\text{pref}(e_1 e_2) = \text{pref}(e_1) + e_1 \text{pref}(e_2)$
 $\text{pref}(e^*) = e^* \text{pref}(e)$ (voir tableau)
- ⑥ Maintenant il faut démontrer que, en fait, $L(\text{pref}(e)) = \text{Pref}(L)$.
- ⑦ ... par induction structurelle, encore (sur tableau).

A vous de jouer

- ① Sur alphabet $\Sigma = \{a, b, c\}$, donnez des expressions rationnelles pour les langages suivants :
- ① $\{w \in \Sigma^* \mid w \text{ contient au moins trois } a\}$
 - ② $\{w \in \Sigma^* \mid w \text{ contient un nombre pair de } a\}$
 - ③ $\{w \in \Sigma^* \mid w \text{ contient exactement 2 ou 3 fois le symbole } a\}$
 - ④ $\{w \in \Sigma^* \mid w \text{ ne contient pas le facteur } cb\}$
- ② Sur alphabet $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, .\}$, donnez des expressions rationnelles pour les langages suivants :
- ① les entiers positifs en base 10
 - ② les entiers relatifs en base 10
 - ③ les nombres décimaux positifs en base 10
 - ④ les nombres décimaux relatifs en base 10
- ③ Sur alphabet $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, ., E\}$, donnez des expressions rationnelles pour les nombres décimaux relatifs en base 10 en notation scientifique.

The image features a series of concentric circles in various shades of red, creating a tunnel-like effect that draws the eye toward the center. At the very center is a solid dark blue circle. Overlaid on this graphic is the text "That's all Folks!" in a white, elegant cursive script. The text is positioned diagonally, starting from the left side and ending near the top right, with the central blue circle acting as a focal point for the phrase.

That's all Folks!