Langages rationnels

MP/MP* Option info

Motivation

De nombreux linguistes (par exemple Noam Chomsky) ont essayé de classer les langues naturelles (français, anglais...) mathématiquement.

Motivation

De nombreux linguistes (par exemple Noam Chomsky) ont essayé de classer les langues naturelles (français, anglais...) mathématiquement.

Les langues naturelles étant trop compliquées, ces idées ont plutôt servi pour théoriser les langages de programmation.

Définitions

- Un **alphabet** est un ensemble Σ **fini**, dont les éléments sont des **lettres**

Définitions

- Un **alphabet** est un ensemble Σ **fini**, dont les éléments sont des **lettres**
- Un **mot** m d'un alphabet Σ est une suite finie m_1, ..., m_n de lettres de Σ, et on note $m = m_1 ... m_n$
  n est la **longueur** de m, qu'on note $|m|$

Définitions

- Un **alphabet** est un ensemble Σ **fini**, dont les éléments sont des **lettres**
- Un **mot** m d'un alphabet Σ est une suite finie m_1, ..., m_n de lettres de Σ, et on note $m = m_1 ... m_n$
  n est la **longueur** de m, qu'on note $|m|$
- Le **mot vide** (contenant aucune lettre) est noté ε

Définitions

- Un **alphabet** est un ensemble Σ **fini**, dont les éléments sont des **lettres**
- Un **mot** m d'un alphabet Σ est une suite finie m_1, ..., m_n de lettres de Σ, et on note $m = m_1 ... m_n$
  n est la **longueur** de m, qu'on note $|m|$
- Le **mot vide** (contenant aucune lettre) est noté ε
- On note Σ^* l'ensemble des mots de Σ et $\Sigma^+ = \Sigma^* \backslash \{\varepsilon\}$

Définitions

- Un **alphabet** est un ensemble Σ **fini**, dont les éléments sont des **lettres**
- Un **mot** m d'un alphabet Σ est une suite finie m_1, ..., m_n de lettres de Σ, et on note $m = m_1 ... m_n$
  n est la **longueur** de m, qu'on note $|m|$
- Le **mot vide** (contenant aucune lettre) est noté ε
- On note Σ^* l'ensemble des mots de Σ et $\Sigma^+ = \Sigma^* \backslash \{\varepsilon\}$
- On note Σ^n l'ensemble des mots de Σ de taille n.

Définitions

- La **concaténation** de deux mots $u = u_1 ... u_n$ et $v = v_1 ... v_p$ est:

$$uv = u_1 ... u_n v_1 ... v_p$$

Définitions

- La **concaténation** de deux mots $u = u_1 ... u_n$ et $v = v_1 ... v_p$ est:
$$uv = u_1 ... u_n v_1 ... v_p$$
- Si u est un mot, $u^k = \underbrace{uu...u}_{k}$ (et $u^0 = \varepsilon$)

Définitions

- La **concaténation** de deux mots $u = u_1 ... u_n$ et $v = v_1 ... v_p$ est:
$$uv = u_1 ... u_n v_1 ... v_p$$
- Si u est un mot, $u^k = \underbrace{uu ... u}_{k}$ (et $u^0 = \varepsilon$)

Soit m un mot de Σ^*:

- u est un **préfixe** de m s'il existe un mot v tel que $m = uv$
- u est un **suffixe** de m s'il existe un mot v tel que $m = vu$
- u est un **facteur** de m s'il existe des mots v, w tels que $m = vuw$

Définitions

Définition

Un **langage** L sur un alphabet Σ est un ensemble de mots de Σ

Définitions

Définition

Un **langage** L sur un alphabet Σ est un ensemble de mots de Σ, ce qui est équivalent à $L \subseteq \Sigma^*$ ou encore $L \in \mathcal{P}(\Sigma^*)$.

Définitions

Définition

Un **langage** L sur un alphabet Σ est un ensemble de mots de Σ, ce qui est équivalent à $L \subseteq \Sigma^*$ ou encore $L \in \mathcal{P}(\Sigma^*)$.

Exemples:

1. L'ensemble L_1 des mots du dictionnaire français sur $\Sigma = \{a, b, c, ..., z\}$

Définitions

Définition

Un **langage** L sur un alphabet Σ est un ensemble de mots de Σ, ce qui est équivalent à $L \subseteq \Sigma^*$ ou encore $L \in \mathcal{P}(\Sigma^*)$.

Exemples:

1. L'ensemble L_1 des mots du dictionnaire français sur $\Sigma = \{a, b, c, ..., z\}$
2. L'ensemble L_2 des formules arithmétiques sur $\Sigma = \{0, 1, ..., 9, +, -, /, *\}$

Définitions

Définition

Un **langage** L sur un alphabet Σ est un ensemble de mots de Σ, ce qui est équivalent à $L \subseteq \Sigma^*$ ou encore $L \in \mathcal{P}(\Sigma^*)$.

Exemples:

1. L'ensemble L_1 des mots du dictionnaire français sur $\Sigma = \{a, b, c, ..., z\}$
2. L'ensemble L_2 des formules arithmétiques sur $\Sigma = \{0, 1, ..., 9, +, -, /, *\}$
3. L'ensemble L_3 des programmes Caml sur $\Sigma = \{a, ..., z, , !, <, >, ...\}$

Définitions

Définition

Un **langage** L sur un alphabet Σ est un ensemble de mots de Σ, ce qui est équivalent à $L \subseteq \Sigma^*$ ou encore $L \in \mathcal{P}(\Sigma^*)$.

Exemples:

1. L'ensemble L_1 des mots du dictionnaire français sur $\Sigma = \{a, b, c, ..., z\}$
2. L'ensemble L_2 des formules arithmétiques sur $\Sigma = \{0, 1, ..., 9, +, -, /, *\}$
3. L'ensemble L_3 des programmes Caml sur $\Sigma = \{a, ..., z, , !, <, >, ...\}$
4. L'ensemble L_4 des ADN sur $\Sigma = \{A, C, G, T\}$

Mots de Dyck

Le langage D des mots de Dyck est l'ensemble des mots bien parenthésés sur $\Sigma = \{\ (,\ )\ \}$.

Mots de Dyck

Le langage D des mots de Dyck est l'ensemble des mots bien parenthésés sur $\Sigma = \{ \, (, \ ) \, \}$.
Exemples: $()() \in D$, $((())) \in D$ mais $()) \notin D$, $)()) \notin D$.

Mots de Dyck

Le langage D des mots de Dyck est l'ensemble des mots bien parenthésés sur $\Sigma = \{\ (,\ )\ \}$.
Exemples: $()() \in D$, $((())) \in D$ mais $()) \notin D$, $)()) \notin D$.

On note D_n l'ensemble des mots de Dyck de longueur $\boldsymbol{2n}$ et $C_n = |D_n|$.

Mots de Dyck

Le langage D des mots de Dyck est l'ensemble des mots bien parenthésés sur $\Sigma = \{\ (,\ )\ \}$.
Exemples: $()() \in D$, $((())) \in D$ mais $()) \notin D$, $)()) \notin D$.

On note D_n l'ensemble des mots de Dyck de longueur $\boldsymbol{2n}$ et $C_n = |D_n|$.

Théorème (nombre de Catalan)

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Mots de Dyck

Commençons par montrer que:

Théorème

$$C_0 = 1$$

$$\forall n \in \mathbb{N}, C_{n+1} = \sum_{k=0}^{n} C_k C_{n-k}$$

Preuve:

Mots de Dyck

Commençons par montrer que:

Théorème

$$C_0 = 1$$

$$\forall n \in \mathbb{N}, C_{n+1} = \sum_{k=0}^{n} C_k C_{n-k}$$

Preuve: tout mot de Dyck de taille $2(n+1)$ s'écrit de façon **unique** de la forme $(u)v$ où u et v sont des mots de Dyck de tailles $2k$ et $2n-2k$.

Mots de Dyck

Commençons par montrer que:

Théorème

$$C_0 = 1$$

$$\forall n \in \mathbb{N}, C_{n+1} = \sum_{k=0}^{n} C_k C_{n-k}$$

Preuve: tout mot de Dyck de taille $2(n+1)$ s'écrit de façon **unique** de la forme $(u)v$ où u et v sont des mots de Dyck de tailles $2k$ et $2n-2k$. Donc la fonction suivante est une bijection:

$$\begin{array}{ccc} \bigcup_{k<n} D_k \times D_{n-k} & \longrightarrow & D_{n+1} \\ (u, v) & \longmapsto & (u)v \end{array}$$

Théorème (nombre de Catalan)

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Théorème (nombre de Catalan)

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Preuve 1: par récurrence sur n.

Théorème (nombre de Catalan)

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Preuve 1: par récurrence sur n.

Preuve 2: soit $S(x) = \sum_{n=0}^{\infty} C_n x^n$.

Théorème (nombre de Catalan)

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

<u>Preuve 1</u>: par récurrence sur n.

<u>Preuve 2</u>: soit $S(x) = \sum_{n=0}^{\infty} C_n x^n$.Alors:

$$S(x) = 1 + \sum_{n=1}^{\infty}\sum_{k=0}^{n-1} C_k C_{n-1-k} x^n = 1 + x\sum_{n=1}^{\infty}\sum_{k=0}^{n-1} C_k C_{n-1-k} x^{n-1}$$

Théorème (nombre de Catalan)

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Preuve 1: par récurrence sur n.

Preuve 2: soit $S(x) = \sum_{n=0}^{\infty} C_n x^n$.Alors:

$$S(x) = 1 + \sum_{n=1}^{\infty}\sum_{k=0}^{n-1} C_k C_{n-1-k} x^n = 1 + x\sum_{n=1}^{\infty}\sum_{k=0}^{n-1} C_k C_{n-1-k} x^{n-1}$$

$$S(x) = 1 + x\sum_{n=0}^{\infty}\sum_{k=0}^{n} C_k C_{n-k} x^n = 1 + xS(x)^2$$

Théorème (nombre de Catalan)

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

<u>Preuve 1</u>: par récurrence sur n.

<u>Preuve 2</u>: soit $S(x) = \sum_{n=0}^{\infty} C_n x^n$.Alors:

$$S(x) = 1 + \sum_{n=1}^{\infty}\sum_{k=0}^{n-1} C_k C_{n-1-k} x^n = 1 + x\sum_{n=1}^{\infty}\sum_{k=0}^{n-1} C_k C_{n-1-k} x^{n-1}$$

$$S(x) = 1 + x\sum_{n=0}^{\infty}\sum_{k=0}^{n} C_k C_{n-k} x^n = 1 + xS(x)^2$$

On trouve alors $S(x) = \frac{1-\sqrt{1-4x}}{2x}$ que l'on peut développer en série entière et permet de trouver $C_n = \frac{1}{n+1}\binom{2n}{n}$ par unicité des coefficients.

Théorème (nombre de Catalan)

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Théorème (nombre de Catalan)

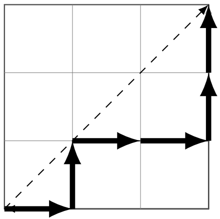
$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Preuve 3: à chaque mot de Dyck on associe un **chemin de Dyck**, qui commence au point (0, 0) et se déplaçant à droite à chaque « (» rencontré et en haut à chaque «) » rencontré.

Théorème (nombre de Catalan)

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

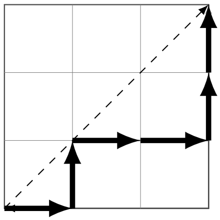
Preuve 3: à chaque mot de Dyck on associe un **chemin de Dyck**, qui commence au point (0, 0) et se déplaçant à droite à chaque « (» rencontré et en haut à chaque «) » rencontré.
Par exemple, le mot ()(()) est associé à:



Théorème (nombre de Catalan)

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Preuve 3: à chaque mot de Dyck on associe un **chemin de Dyck**, qui commence au point (0, 0) et se déplaçant à droite à chaque « (» rencontré et en haut à chaque «) » rencontré.
Par exemple, le mot ()(()) est associé à:



Ceci donne une bijection de D_n vers l'ensemble P_n des chemins de $(0, 0)$ à (n, n) en dessous de la diagonale.

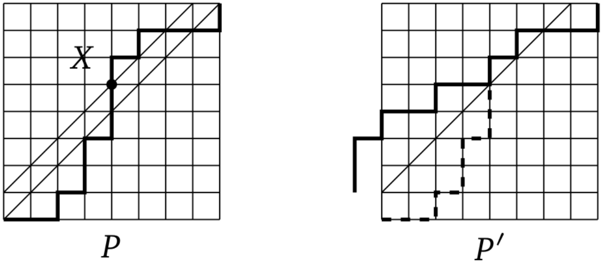
Le nombre de chemins P de $(0,0)$ à (n,n) est

Le nombre de chemins P de $(0,0)$ à (n,n) est $\binom{2n}{n}$.

Soit $P \notin P_n$: notons X le premier point de P au dessus de la diagonale. On applique la symétrie d'équation $y = x + 1$ sur la partie de P de $(0,0)$ à X. On obtient un nouveau chemin P'.

Le nombre de chemins P de $(0,0)$ à (n,n) est $\binom{2n}{n}$.

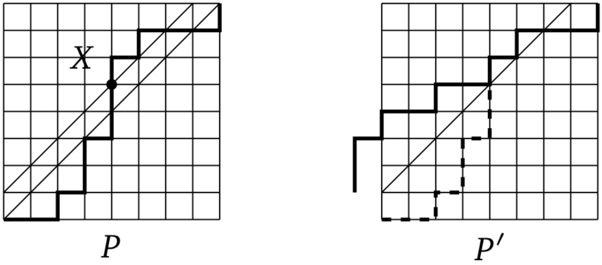
Soit $P \notin P_n$: notons X le premier point de P au dessus de la diagonale. On applique la symétrie d'équation $y = x + 1$ sur la partie de P de $(0,0)$ à X. On obtient un nouveau chemin P'.



$P \longmapsto P'$ est une bijection de $\overline{P_n}$ vers l'ensemble des chemins de $(-1,1)$ à (n,n)

Le nombre de chemins P de $(0,0)$ à (n,n) est $\binom{2n}{n}$.

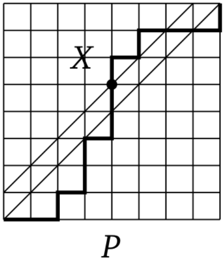
Soit $P \notin P_n$: notons X le premier point de P au dessus de la diagonale. On applique la symétrie d'équation $y = x + 1$ sur la partie de P de $(0,0)$ à X. On obtient un nouveau chemin P'.

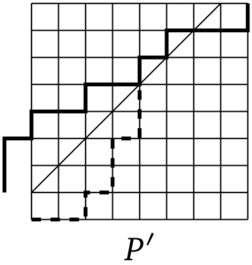


$P \longmapsto P'$ est une bijection de $\overline{P_n}$ vers l'ensemble des chemins de $(-1,1)$ à (n,n) (preuve: on peut trouver sa fonction inverse).

Le nombre de chemins P de $(0,0)$ à (n,n) est $\binom{2n}{n}$.

Soit $P \notin P_n$: notons X le premier point de P au dessus de la diagonale. On applique la symétrie d'équation $y = x + 1$ sur la partie de P de $(0,0)$ à X. On obtient un nouveau chemin P'.





$P \longmapsto P'$ est une bijection de $\overline{P_n}$ vers l'ensemble des chemins de $(-1,1)$ à (n,n) (preuve: on peut trouver sa fonction inverse).

Donc $|P_n| = \binom{2n}{n} - \binom{2n}{n-1} = ... = \boxed{\frac{1}{n+1}\binom{2n}{n}}$

On en déduit aussi que:

Théorème

Le nombre d'arbres binaires à n sommets (sans considérer les étiquettes) est $\dfrac{1}{n+1}\dbinom{2n}{n}$

On en déduit aussi que:

Théorème

Le nombre d'arbres binaires à n sommets (sans considérer les étiquettes) est $\frac{1}{n+1}\binom{2n}{n}$

Preuve 1: même équation de récurrence que les mots de Dyck.

On en déduit aussi que:

Théorème

Le nombre d'arbres binaires à n sommets (sans considérer les étiquettes) est $\dfrac{1}{n+1}\dbinom{2n}{n}$

Preuve 1: même équation de récurrence que les mots de Dyck.
Preuve 2: la fonction suivante est une bijection de l'ensemble des arbres binaires à n sommets vers l'ensemble des mots de Dyck de longueur $2n$:

$$f(N(g,d)) = (f(g))f(d)$$

$$f(V) = \varepsilon$$

Problèmes classiques

On souhaiterait des algorithmes pour résoudre les problèmes suivants:

Problème

Étant donné un langage L et un mot m, est-ce que $m \in L$?

Applications: correcteur orthographique, coloration syntaxique...

Problèmes classiques

On souhaiterait des algorithmes pour résoudre les problèmes suivants:

Problème

Étant donné un langage L et un mot m, est-ce que $m \in L$?

Applications: correcteur orthographique, coloration syntaxique...

Problème

Étant donné un texte s et un langage L, s contient-il un mot de L?

Problèmes classiques

On souhaiterait des algorithmes pour résoudre les problèmes suivants:

Problème

Étant donné un langage L et un mot m, est-ce que $m \in L$?

Applications: correcteur orthographique, coloration syntaxique...

Problème

Étant donné un texte s et un langage L, s contient-il un mot de L?
(Cas particulier $L = \{m\}$: le mot m apparaît-il dans un texte s?)

Application: recherche de motif (adresse mail, séquence d'ADN...) dans un texte.

Cardinalité

Soit Σ un alphabet non vide.

Théorème

Σ^+ (et donc aussi Σ^*) est infini dénombrable.

Cardinalité

Soit Σ un alphabet non vide.

Théorème

Σ^+ (et donc aussi Σ^*) est infini dénombrable.

Preuve: on identifie les p lettres de Σ avec les entiers de 0 à $p-1$. L'écriture en base p donne une bijection de $\mathbb{N}$ vers Σ^+.

Cardinalité

Soit Σ un alphabet non vide.

Théorème

Σ^+ (et donc aussi Σ^*) est infini dénombrable.

Preuve: on identifie les p lettres de Σ avec les entiers de 0 à $p-1$.
L'écriture en base p donne une bijection de $\mathbb{N}$ vers Σ^+.

On en déduit:

Théorème

Un langage $L \subseteq \Sigma^*$ est au plus dénombrable.

Par exemple, le langage des programmes Caml est dénombrable.

Théorème de Cantor

Un ensemble E ne peut pas être en bijection avec $\mathcal{P}(E)$.

Théorème de Cantor

Un ensemble E ne peut pas être en bijection avec $\mathcal{P}(E)$.

<u>Preuve</u>: si $f : E \longrightarrow \mathcal{P}(E)$ alors $Y = \{x \in E \mid x \notin f(x)\}$ n'a pas d'antécédent par f.

Théorème de Cantor

Un ensemble E ne peut pas être en bijection avec $\mathcal{P}(E)$.

Preuve: si $f : E \longrightarrow \mathcal{P}(E)$ alors $Y = \{x \in E \mid x \notin f(x)\}$ n'a pas d'antécédent par f.

Corollaire

L'ensemble $\mathcal{P}(\Sigma^*)$ des langages sur Σ n'est pas dénombrable.

Théorème de Cantor

Un ensemble E ne peut pas être en bijection avec $\mathcal{P}(E)$.

Preuve: si $f : E \longrightarrow \mathcal{P}(E)$ alors $Y = \{x \in E \mid x \notin f(x)\}$ n'a pas d'antécédent par f.

Corollaire

L'ensemble $\mathcal{P}(\Sigma^*)$ des langages sur Σ n'est pas dénombrable.

Comme l'ensemble des langages est indénombrable alors que l'ensemble des programmes Caml est dénombrable, il existe des langages L pour lesquels le problème suivant ne peut pas être résolu par un algorithme:

Problème

Étant donné un mot m, est-ce que $m \in L$?

Théorème de Cantor

Un ensemble E ne peut pas être en bijection avec $\mathcal{P}(E)$.

Preuve: si $f : E \longrightarrow \mathcal{P}(E)$ alors $Y = \{x \in E \mid x \notin f(x)\}$ n'a pas d'antécédent par f.

Corollaire

L'ensemble $\mathcal{P}(\Sigma^*)$ des langages sur Σ n'est pas dénombrable.

Comme l'ensemble des langages est indénombrable alors que l'ensemble des programmes Caml est dénombrable, il existe des langages L pour lesquels le problème suivant ne peut pas être résolu par un algorithme:

Problème

Étant donné un mot m, est-ce que $m \in L$?

On va donc se restreindre à un ensemble plus simple de langages

Opérations

On définit les opérations suivantes sur des langages de même alphabet:

- **concaténation**: $L_1 L_2 = \{m_1 m_2 \mid m_1 \in L_1,\ m_2 \in L_2\}$
  (ne pas confondre L^2 avec $\{mm \mid m \in L\}$)

Opérations

On définit les opérations suivantes sur des langages de même alphabet:

- **concaténation**: $L_1L_2 = \{m_1m_2 \mid m_1 \in L_1,\ m_2 \in L_2\}$
  (ne pas confondre L^2 avec $\{mm \mid m \in L\}$)
- **puissance**: $L^0 = \{\varepsilon\}$ et $L^n = LL^{n-1}$ (mots obtenus comme concaténation de n mots de L)

Opérations

On définit les opérations suivantes sur des langages de même alphabet:

- **concaténation**: $L_1 L_2 = \{m_1 m_2 \mid m_1 \in L_1,\ m_2 \in L_2\}$
  (ne pas confondre L^2 avec $\{mm \mid m \in L\}$)
- **puissance**: $L^0 = \{\varepsilon\}$ et $L^n = LL^{n-1}$ (mots obtenus comme concaténation de n mots de L)
- **étoile de Kleene**: $L^* = \bigcup_{k \in \mathbb{N}} L^k$ (mots obtenus comme répétition d'un nombre quelconque de mots de L)

Opérations

On définit les opérations suivantes sur des langages de même alphabet:

- **concaténation**: $L_1L_2 = \{m_1m_2 \mid m_1 \in L_1,\ m_2 \in L_2\}$
  (ne pas confondre L^2 avec $\{mm \mid m \in L\}$)
- **puissance**: $L^0 = \{\varepsilon\}$ et $L^n = LL^{n-1}$ (mots obtenus comme concaténation de n mots de L)
- **étoile de Kleene**: $L^* = \bigcup_{k \in \mathbb{N}} L^k$ (mots obtenus comme répétition d'un nombre quelconque de mots de L)
- **opérations ensemblistes**: $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 \backslash L_2$...

Langage rationnel

Langages rationnels

L'ensemble des **langages rationnels** de Σ est le plus petit ensemble de langages de Σ **contenant les langages finis** de Σ et stable par **concaténation**, **union**, **étoile de Kleene**.

Langage rationnel

Langages rationnels

L'ensemble des **langages rationnels** de Σ est le plus petit ensemble de langages de Σ **contenant les langages finis** de Σ et stable par **concaténation**, **union**, **étoile de Kleene**.

Exemples:

1. Soit m un mot. Alors $\{m\}$ est un langage rationnel, qu'on note aussi simplement m.

Langage rationnel

Langages rationnels

L'ensemble des **langages rationnels** de Σ est le plus petit ensemble de langages de Σ **contenant les langages finis** de Σ et stable par **concaténation**, **union**, **étoile de Kleene**.

Exemples:

1. Soit m un mot. Alors $\{m\}$ est un langage rationnel, qu'on note aussi simplement m.
2. Σ est un langage rationnel donc Σ^* aussi.

Langage rationnel

Langages rationnels

L'ensemble des **langages rationnels** de Σ est le plus petit ensemble de langages de Σ **contenant les langages finis** de Σ et stable par **concaténation**, **union**, **étoile de Kleene**.

Exemples:

1. Soit m un mot. Alors $\{m\}$ est un langage rationnel, qu'on note aussi simplement m.
2. Σ est un langage rationnel donc Σ^* aussi.
3. Soit m un mot. L'ensemble des mots ayant comme facteur m

Langage rationnel

Langages rationnels

L'ensemble des **langages rationnels** de Σ est le plus petit ensemble de langages de Σ **contenant les langages finis** de Σ et stable par **concaténation**, **union**, **étoile de Kleene**.

Exemples:

1. Soit m un mot. Alors $\{m\}$ est un langage rationnel, qu'on note aussi simplement m.
2. Σ est un langage rationnel donc Σ^* aussi.
3. Soit m un mot. L'ensemble des mots ayant comme facteur m est égal à $\Sigma^* m \Sigma^*$: c'est donc un langage rationnel.
4. L'ensemble des mots de taille pair est égal à

Langage rationnel

Langages rationnels

L'ensemble des **langages rationnels** de Σ est le plus petit ensemble de langages de Σ **contenant les langages finis** de Σ et stable par **concaténation**, **union**, **étoile de Kleene**.

Exemples:

1. Soit m un mot. Alors $\{m\}$ est un langage rationnel, qu'on note aussi simplement m.
2. Σ est un langage rationnel donc Σ^* aussi.
3. Soit m un mot. L'ensemble des mots ayant comme facteur m est égal à $\Sigma^* m \Sigma^*$: c'est donc un langage rationnel.
4. L'ensemble des mots de taille pair est égal à $(\Sigma^2)^*$: c'est donc un langage rationnel.

Langage rationnel

Dit autrement:

Langages rationnels

- Tout langage fini est rationnel
- L_1 et L_2 rationnels $\implies$ $L_1 \cup L_2$ rationnel
- L_1 et L_2 rationnels $\implies$ $L_1 L_2$ rationnel
- L rationnel $\implies$ L^* rationnel

Attention: une union infinie de langages rationnels n'est pas forcément rationnel.

On peut démontrer une propriété $\mathcal{P}(L)$ sur les langages rationnels L par induction structurelle, en montrant:

- $\mathcal{P}(L)$ est vraie pour les langages L finis (cas de base)
- $\mathcal{P}(L_1) \wedge \mathcal{P}(L_2) \implies \mathcal{P}(L_1 L_2)$
- $\mathcal{P}(L_1) \wedge \mathcal{P}(L_2) \implies \mathcal{P}(L_1 \cup L_2)$
- $\mathcal{P}(L) \implies \mathcal{P}(L^*)$

On peut démontrer une propriété $\mathcal{P}(L)$ sur les langages rationnels L par induction structurelle, en montrant:

- $\mathcal{P}(L)$ est vraie pour les langages L finis (cas de base)
- $\mathcal{P}(L_1) \wedge \mathcal{P}(L_2) \implies \mathcal{P}(L_1 L_2)$
- $\mathcal{P}(L_1) \wedge \mathcal{P}(L_2) \implies \mathcal{P}(L_1 \cup L_2)$
- $\mathcal{P}(L) \implies \mathcal{P}(L^*)$

L'ensemble des langages L vérifiant $\mathcal{P}(L)$ contient alors les langages finis et est stable par concaténation, union, étoile de Kleene.
Donc il contient tous les langages rationnels, par définition.

On peut démontrer une propriété $\mathcal{P}(L)$ sur les langages rationnels L par induction structurelle, en montrant:

- $\mathcal{P}(L)$ est vraie pour les langages L finis (cas de base)
- $\mathcal{P}(L_1) \wedge \mathcal{P}(L_2) \implies \mathcal{P}(L_1 L_2)$
- $\mathcal{P}(L_1) \wedge \mathcal{P}(L_2) \implies \mathcal{P}(L_1 \cup L_2)$
- $\mathcal{P}(L) \implies \mathcal{P}(L^*)$

L'ensemble des langages L vérifiant $\mathcal{P}(L)$ contient alors les langages finis et est stable par concaténation, union, étoile de Kleene.
Donc il contient tous les langages rationnels, par définition.

Exercice

Soit u un mot et L un langage rationnel sur Σ.
Montrer que $u^{-1}L = \{v \in \Sigma^* \mid uv \in L\}$ est rationnel.

Les expressions rationnelles sont une notation un peu plus compacte pour représenter un langage rationnel.

Expressions rationnelles

L'ensemble des **expressions rationnelles** sur un alphabet Σ est le plus petit langage $\mathcal{R}$ sur $\Sigma \cup \{\emptyset, \varepsilon, +, ^*, (,)\}$ vérifiant:

- $\forall a \in \Sigma$, $a \in \mathcal{R}$
- $\emptyset \in \mathcal{R}$, $\varepsilon \in \mathcal{R}$
- $\forall e, e' \in \mathcal{R}$, $(e + e') \in \mathcal{R}$ et $(ee') \in \mathcal{R}$
- $\forall e \in \mathcal{R}$, $e^* \in \mathcal{R}$

Les expressions rationnelles sont une notation un peu plus compacte pour représenter un langage rationnel.

Expressions rationnelles

L'ensemble des **expressions rationnelles** sur un alphabet Σ est le plus petit langage $\mathcal{R}$ sur $\Sigma \cup \{\emptyset, \varepsilon, +, {}^*, (,)\}$ vérifiant:

- $\forall a \in \Sigma$, $a \in \mathcal{R}$
- $\emptyset \in \mathcal{R}$, $\varepsilon \in \mathcal{R}$
- $\forall e, e' \in \mathcal{R}$, $(e + e') \in \mathcal{R}$ et $(ee') \in \mathcal{R}$
- $\forall e \in \mathcal{R}$, $e^* \in \mathcal{R}$

En pratique, on utilisera seulement les parenthèses nécessaires.
Exemple: $(((ab)c) + d)$ sera noté $abc + d$.

Les expressions rationnelles sont une notation un peu plus compacte pour représenter un langage rationnel.

Expressions rationnelles

L'ensemble des **expressions rationnelles** sur un alphabet Σ est le plus petit langage $\mathcal{R}$ sur $\Sigma \cup \{\emptyset, \varepsilon, +, ^*, (,)\}$ vérifiant:

- $\forall a \in \Sigma$, $a \in \mathcal{R}$
- $\emptyset \in \mathcal{R}$, $\varepsilon \in \mathcal{R}$
- $\forall e, e' \in \mathcal{R}$, $(e + e') \in \mathcal{R}$ et $(ee') \in \mathcal{R}$
- $\forall e \in \mathcal{R}$, $e^* \in \mathcal{R}$

En pratique, on utilisera seulement les parenthèses nécessaires.
Exemple: $(((ab)c) + d)$ sera noté $abc + d$.

Si $\Sigma = \{a, b\}$: a^*, $(a + aba)^*$, $a(a + b)^*b$ sont des expressions rationnelles.

Expressions rationnelles

L'ensemble des **expressions rationnelles** sur un alphabet Σ est le plus petit langage $\mathcal{R}$ sur $\Sigma \cup \{\emptyset, \varepsilon, +, ^*, (,)\}$ vérifiant:

- $\forall a \in \Sigma,\ a \in \mathcal{R}$
- $\emptyset \in \mathcal{R},\ \varepsilon \in \mathcal{R}$
- $\forall e, e' \in \mathcal{R},\ (e + e') \in \mathcal{R}$ et $ee' \in \mathcal{R}$
- $\forall e \in \mathcal{R},\ e^* \in \mathcal{R}$

```
type 'a regexp =
    Epsilon
  | Vide
  | L of 'a
  | Somme of 'a regexp * 'a regexp
  | Concat of 'a regexp * 'a regexp
  | Etoile of 'a regexp;;
```

Expressions rationnelles

L'ensemble des **expressions rationnelles** sur un alphabet Σ est le plus petit langage $\mathcal{R}$ sur $\Sigma \cup \{\emptyset, \varepsilon, +, ^*, (,)\}$ vérifiant:

- $\forall a \in \Sigma,\ a \in \mathcal{R}$
- $\emptyset \in \mathcal{R},\ \varepsilon \in \mathcal{R}$
- $\forall e, e' \in \mathcal{R},\ (e + e') \in \mathcal{R}$ et $ee' \in \mathcal{R}$
- $\forall e \in \mathcal{R},\ e^* \in \mathcal{R}$

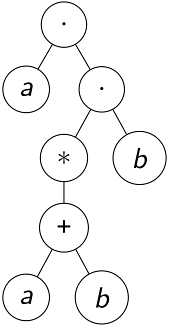
```
type 'a regexp =
    Epsilon
  | Vide
  | L of 'a
  | Somme of 'a regexp * 'a regexp
  | Concat of 'a regexp * 'a regexp
  | Etoile of 'a regexp;;
```

Par exemple, $a(a+b)^*b$ est représenté par:

```
Concat(L("a"), Concat(Etoile(Somme(L("a"), L("b"))), L("b")))
```

```
type 'a regexp =
    Epsilon
  | Vide
  | L of 'a
  | Somme of 'a regexp * 'a regexp
  | Concat of 'a regexp * 'a regexp
  | Etoile of 'a regexp;;
```

On peut aussi représenter une expression rationnelle par un arbre.
Par exemple, $a(a+b)^*b$ peut être représenté par:



Langage d'une expression rationnelle

Le langage $L(e)$ d'une expression rationnelle e est définie récursivement:

- $L(a) = \{a\}$ si $a \in \Sigma$
- $L(\emptyset) = \emptyset$, $L(\varepsilon) = \{\varepsilon\}$
- $L(e + e') = L(e) \cup L(e')$
- $L(ee') = L(e)L(e')$
- $L(e^*) = L(e)^*$

Langage d'une expression rationnelle

Le langage $L(e)$ d'une expression rationnelle e est définie récursivement:

- $L(a) = \{a\}$ si $a \in \Sigma$
- $L(\emptyset) = \emptyset$, $L(\varepsilon) = \{\varepsilon\}$
- $L(e + e') = L(e) \cup L(e')$
- $L(ee') = L(e)L(e')$
- $L(e^*) = L(e)^*$

En pratique on n'utilise pas cette définition, mais on déduit le langage d'une expression rationnelle en interprétant $+$ par $\cup$ (« ou »).

Langage d'une expression rationnelle

Le langage $L(e)$ d'une expression rationnelle e est définie récursivement:

- $L(a) = \{a\}$ si $a \in \Sigma$
- $L(\emptyset) = \emptyset$, $L(\varepsilon) = \{\varepsilon\}$
- $L(e + e') = L(e) \cup L(e')$
- $L(ee') = L(e)L(e')$
- $L(e^*) = L(e)^*$

Exemples avec $\Sigma = \{a, b\}$:

- $L((a + b)^*)$ est

Langage d'une expression rationnelle

Le langage $L(e)$ d'une expression rationnelle e est définie récursivement:

- $L(a) = \{a\}$ si $a \in \Sigma$
- $L(\emptyset) = \emptyset$, $L(\varepsilon) = \{\varepsilon\}$
- $L(e + e') = L(e) \cup L(e')$
- $L(ee') = L(e)L(e')$
- $L(e^*) = L(e)^*$

Exemples avec $\Sigma = \{a, b\}$:

- $L((a + b)^*)$ est l'ensemble de tous les mots Σ^*.
- $L(a(a + b)^* b)$ est

Langage d'une expression rationnelle

Le langage $L(e)$ d'une expression rationnelle e est définie récursivement:

- $L(a) = \{a\}$ si $a \in \Sigma$
- $L(\emptyset) = \emptyset$, $L(\varepsilon) = \{\varepsilon\}$
- $L(e + e') = L(e) \cup L(e')$
- $L(ee') = L(e)L(e')$
- $L(e^*) = L(e)^*$

Exemples avec $\Sigma = \{a, b\}$:

- $L((a + b)^*)$ est l'ensemble de tous les mots Σ^*.
- $L(a(a + b)^*b)$ est l'ensemble des mots commençant par la lettre a et finissant par b.
- $L(a(a + b)^* + b(a + b)^* + \varepsilon)$ est

Langage d'une expression rationnelle

Le langage $L(e)$ d'une expression rationnelle e est définie récursivement:

- $L(a) = \{a\}$ si $a \in \Sigma$
- $L(\emptyset) = \emptyset$, $L(\varepsilon) = \{\varepsilon\}$
- $L(e + e') = L(e) \cup L(e')$
- $L(ee') = L(e)L(e')$
- $L(e^*) = L(e)^*$

Exemples avec $\Sigma = \{a, b\}$:

- $L((a + b)^*)$ est l'ensemble de tous les mots Σ^*.
- $L(a(a + b)^* b)$ est l'ensemble des mots commençant par la lettre a et finissant par b.
- $L(a(a + b)^* + b(a + b)^* + \varepsilon)$ est aussi Σ^*.

Des expressions régulières différentes peuvent donc définir le même langage.

En notant $e_1 \equiv e_2 \iff L(e_1) = L(e_2)$:

Propriétés sur les expressions rationnelles

- $\emptyset e \equiv e\emptyset \equiv \emptyset$
- $\varepsilon e \equiv e\varepsilon \equiv e$
- $(e_1 + e_2)e_3 \equiv e_1 e_3 + e_2 e_3$ (distributivité)
- $e_1(e_2 e_3) \equiv (e_1 e_2)e_3$ (associativité)

Les expressions rationnelles donnent une définition équivalente des langages rationnels:

Langages rationnels

Un langage L est rationnel

$$\iff$$

Il existe une expression rationnelle e telle que $L = L(e)$

Exemples:

- langage sur $\Sigma = \{a, b\}$ contenant un nombre pair de a:

Les expressions rationnelles donnent une définition équivalente des langages rationnels:

Langages rationnels

Un langage L est rationnel

$$\Longleftrightarrow$$

Il existe une expression rationnelle e telle que $L = L(e)$

Exemples:

- langage sur $\Sigma = \{a, b\}$ contenant un nombre pair de a: $L((ab^*a + b)^*)$
- langage sur $\Sigma = \{a, b\}$ contenant un nombre impair de a:

Les expressions rationnelles donnent une définition équivalente des langages rationnels:

Langages rationnels

Un langage L est rationnel

$$\iff$$

Il existe une expression rationnelle e telle que $L = L(e)$

Exemples:

- langage sur $\Sigma = \{a, b\}$ contenant un nombre pair de a: $L((ab^*a + b)^*)$
- langage sur $\Sigma = \{a, b\}$ contenant un nombre impair de a: $L(b^*a(ab^*a + b)^*)$

Langage non rationnel

Soit Σ un alphabet non vide.

Théorème

L'ensemble des langages rationnels sur Σ est infini dénombrable.

Preuve:

Langage non rationnel

Soit Σ un alphabet non vide.

Théorème

L'ensemble des langages rationnels sur Σ est infini dénombrable.

Preuve: l'ensemble des expressions rationnelles sur Σ est dénombrable (c'est un langage) donc l'ensemble des langages rationnels aussi.

Langage non rationnel

Soit Σ un alphabet non vide.

Théorème

L'ensemble des langages rationnels sur Σ est infini dénombrable.

Preuve: l'ensemble des expressions rationnelles sur Σ est dénombrable (c'est un langage) donc l'ensemble des langages rationnels aussi.

Comme l'ensemble de tous les langages sur Σ est non dénombrable:

Corollaire

Il existe des langages non rationnels sur Σ.

Langage non rationnel

Soit Σ un alphabet non vide.

Théorème

L'ensemble des langages rationnels sur Σ est infini dénombrable.

Preuve: l'ensemble des expressions rationnelles sur Σ est dénombrable (c'est un langage) donc l'ensemble des langages rationnels aussi.

Comme l'ensemble de tous les langages sur Σ est non dénombrable:

Corollaire

Il existe des langages non rationnels sur Σ.

On verra plus tard comment montrer qu'un langage **n'est pas** rationnel...